RSLinx
Training Guide

Contacting Rockwell Software

Technical Support Telephone—440-646-5800
Technical Support Fax—440-646-5801
World Wide Web—www.software.rockwell.com

Copyright Notice

Trademark Notices

Warranty

This Rockwell Software product is warranted in accord with the product license. The product's performance will be affected by system configuration, the application being performed, operator control and other related factors.

The product's implementation may vary among users.

This manual is as up-to-date as possible at the time of printing; however, the accompanying software may have changed since that time. Rockwell Software reserves the right to change any information contained in this manual or the software at anytime without prior notice.

The instructions in this manual do not claim to cover all the details or variations in the equipment, procedure, or process described, or to provide directions for meeting every possible contingency during installation, operation, or maintenance.

Revision Date: 2/27/2002
Covers: RSLinx 2.20

RSLinx
Training Guide

Table Of Contents

Goals:

By the end of the training period, using this training guide, classroom instruction and experience with RSLinx, you should understand and correctly perform configuration steps for the following:

- The features of RSLinx
- Installation requirements and how to install RSLinx
- How to configure communications hardware in RSLinx
- Using the new RSWho
- How to use the diagnostics in RSLinx
- DDE (Dynamic Data Exchange) and OPC (OLE for Process Control) and how they work
- How to configure DDE/OPC topics
- How to set-up a DDE hot link to Microsoft Excel and make a live graph of the data
- How to write a macro in Microsoft Excel to write data via DDE
- Network DDE - exchanging data between RSLinx server and client applications

Training Section 1: Introduction

Product Description

RSLinx is a complete 32-bit product family that provides communications to a wide range of applications. RSLinx supports Rockwell Software and Allen-Bradley programming, HMI, and component software products.

It serves shop floor data to commercially available DDE client applications, such as Microsoft Excel and Access. This allows real-time data from the plant floor to be brought into applications for display, logging, or trending.

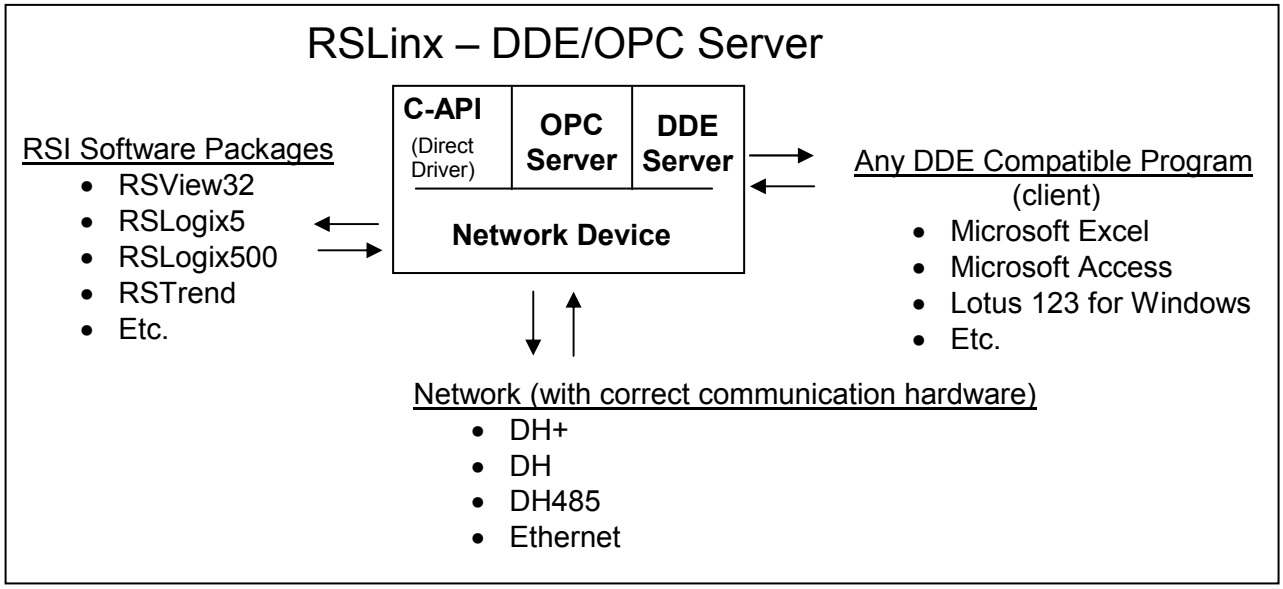
With RSLinx, you can set individual parameters or download recipes to supported devices from a supervisory computer. RSLinx also supports many popular industrial applications developed by OEMs (Original Equipment Manufacturers), VARs (Value-Added Resellers - part of Rockwell Software's Strategic Provider Program), and independent software vendors using the INTERCHANGE Compatible RSLinx 'C' API.

RSLinx is available in packages to suit a variety of needs for both features and functionality. RSLinx OEM is available to act as a communications engine for many of Rockwell Software's HMI software products (RSTrend, WINtelligent VIEW, WINtelligent QUALITY, WINtelligent RECIPE), as well as any third-party product that was developed to use the RSLinx 'C' API.

RSLinx Professional starts with the functionality of RSLinx OEM and adds DDE communications to all products that can act as DDE clients. RSLinx SDK is the Software Development Kit used for creating custom applications that can utilize the communications capabilities of RSLinx and RSLinx OEM.

Its 'C' Application Programming Interface (API) includes over 50 of the most powerful data access and conversion functions of the INTERCHANGE API, plus a few additional functions specific to the RSLinx architecture.

Developed from top to bottom as a true 32-bit application, RSLinx takes full advantage of the multi-threading, multi-tasking, and multi-processing capabilities of the Windows operating systems. RSLinx can run any combination of the above applications simultaneously, through the same or different communication interfaces.



RSLinx can transfer data to and from Allen-Bradley PLCs using common Allen-Bradley communications devices such as the PCMK, 1784-KT board or the 1770-KF2B module.

It supports the Allen-Bradley SLC-500, PLC-2, PLC-3, PLC-5, and PLC-5/250 families of processors all in one package.

It is a complete communications software package, not requiring any external driver software to operate - it is not a TSR (memory resident). It does all of the communications work for commercially available DDE compatible Windows programs as well as Windows programs that a user may design.

With RSLinx and a DDE compatible Windows application such as Excel, a user can install, connect to a PLC, and be on-line collecting data in about 5 minutes.

In some ways, RSLinx is three drivers in one:

- It includes direct driver level access between the PLC and other Rockwell Software products.
- It provides DDE (Dynamic Data Exchange) access to any DDE compatible software program.
- In addition, it now supports OLE for Process Control (OPC).
  - For more information on OPC, or to download the OPC specification, got to:

`http://www.opcfoundation.org`

OPC FOUNDATION

New Features

RSLinx version 2.1 Features

- **Network Shortcuts** – Provides quick access to specific network hiding routing complexities from client applications.
- **New Driver Configuration**
  - Selectively enable/disable drivers
  - Have multiple COM port drivers configured but disabled. Quickly switch between drivers for the same COM port.
  - Specify driver startup state (i.e. Automatic, Manual, Disabled, On Demand)
    - The On Demand selection is only available for serial and Ethernet drivers, and for the PCMK driver only in the Windows 95/98 environment.
  - Only applies to specific drivers (i.e. PCMK on Win 9x, RS-232 serial driver)
    - RS-232 DF1 driver can now use up to 32 COM ports.
- **PLC/SLC Data Monitor** (read only) – New RSLinx Data Monitor ActiveX control provides ability to monitor multiple data files for PLC or SLC devices.
  - Unlike the Data Monitor found in RSLinx 2.0, they now support the RSLogix Emulate 5/500 emulator software!
- **OPC/DDE for DeviceNet** – Read / Write data values to any device on DeviceNet. Use Electronic Data Sheet (EDS) information to access device attributes in a user-friendly format while using Copy to Clipboard.
  - EDS parameters for DeviceNet topics and tags for topics with an associated tag name file. To use the feature, refer to the Copy a DDE link help topic. You can also use the OPC browse feature to get EDS parameters from browse-capable OPC clients.
- **Configuration Backup/Restore** – Save RSLinx configuration information including drivers and DDE/OPC Topics. Backup files can be archived on a network server and retrieved during a restore operation. Included in this functionality is a command line driven interface for Batch file operations.
- **Network Properties** - Users can now specify network polling parameter and address ranges for configured drivers. This is required for offlink network browsing (e.g. KE-KE bridging).

New Features (cont.)

- **Drivers**
  - 1784-PCC PCMCIA for ControlNet (NT driver).
    - The NT kernel mode driver is shipped with the 1784-PCC card (Series B or a reflashed Series A) and must be installed prior to configuring the 1784-PCC driver in RSLinx.
  - 5136-SD/SD2 S&S DH+ (Windows 95/98 driver)
  - DF1 Slave driver
  - DeviceNet 1747-SDN Pass Through driver
  - Support for DeviceNet drivers (1784-PCD, 1784-PCID(S)).
    - Install the driver provided with the hardware.

- **DDE/OPC server diagnostics– Added the following diagnostics:**
  - Handled unsolicited messages
  - Unhandled unsolicited messages
  - DDE Client Connections
  - DDE Client Disconnections
  - OPC Client Connections
  - OPC Client Disconnections

- **New DDE/OPC item modifier** noopt allows the client rather than RSLinx to optimize items.
  - For example, n7:0,L122,noopt ignores the 100 word limit on a DH+ network and reads the full 122 words. However, this item is sent in its own packet.
- **Support for RSLogix 5 and RSLogix 500 data files** (.rss and .rsp) directly for Tag Name Service support.
  - To use the RSLogix 5 and 500 tag database for DDE or OPC, select DDE/OPC > Topic Configuration > Data Collection, enable Use Symbols and select the proper file.
  - DDE and OPC will now use your tag names for PLC5 or SLC addresses. To view what tags RSLinx reads from your data file, select Edit > Copy DDE Link to Clipboard, select the topic, and choose Data Table Address.
- **MicroLogix 1500** processor support.

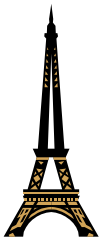
New Features (cont.)

- **RSLinx runs "headless"** when automatically started by RSView32, OPC clients, RSLogix products, and custom C applications.

  - No splash screen appears and the RSLinx main window is not shown. In this case, the RSLinx application is said to be running "headless."
  - RSLinx does however appear as a system tray icon, just as if it were running as a service.
  - To view the existing RSLinx, click the icon, right-click the icon and select Restore, or from the Start menu select Programs > Rockwell Software > RSLinx > RSLinx. Any of these actions will bring the running RSLinx to the foreground.
  - To shut down RSLinx, right-click the icon and select Shutdown, or right-click the icon, select Restore, and choose File > Exit and Shutdown.

RSLinx Lite, RSLinx OEM, and RSLinx Professional are available in the following languages:

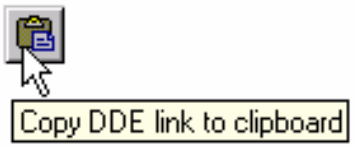
- **English**
- **Spanish**
- **German**
- **French**
- **Italian**
- **Portuguese**



RSLinx Features

RSLinx has had the following existing features since version 2.0 and earlier:

- **Support for all A-B PLC types**; SLC-500 family, PLC-2, PLC-3, PLC-5, PLC-5/250 in one software package.
- **All communications hardware devices** for Allen Bradley PLCs are supported in a single software package. There is no need to buy a separate software package for each hardware device you wish to use and there are no TSR (memory resident) programs to install.
- **Multiple communication hardware devices** may be used at the same time, in the same personal computer. These devices may communicate on one or more Data Highway Pluses and/or Data Highway 485s.
- **Graphical RSWho** function may be configured to present a graphical representation of the active devices on an Allen-Bradley network. It is able to display RSWhos for different highways at the same time and can be set to refresh automatically or manually.
- Complete **diagnostics and error reporting**. It also includes diagnostics for DDE links that may be active.
- **Individual read and writes** (word/bits can be read and/or written).
- **Block reads and writes** (not supported when using the emulators).
- **Unsolicited messages** with Allen-Bradley PLCs. (not supported when using the emulators).
- Copy Link to **Windows Clipboard** supported to ease establishment of DDE links.



- Automatic **optimization** of data communications.
- **Local and Remote stations** through 1785-KA\1785-KA5\5130-KA module bridges.
- Network support via **NET DDE**.
- **Interrupt support** in the 1784-KT drivers for improved throughput.

RSLinx Processor/Communications Device Support Matrixes

Allen-Bradley Hardware	Ethernet	DH+	DH485	DF1 (Serial)	DeviceNet	ControlNet
1747-PIC			X			
1770-KFC						X
1770-KFD (Serial Interface)					X	
1770-KF2		X				
1770-KF3/1747-KE			X			
1771-SDNPT *					X	
1784-KT/KTX/PCMK		X	X			
1784-KTC(X) or 9904-KTCX						X
1784-PCC (PCMCIA CARD)						X
1784-PCD (PCMCIA CARD)					X	
1784-PCDIS (PCI CARD)					X (NT only)	
1784-PCMK		X	X			
1784-PKTX		X	X			
1785-KE		X				
AB Ethernet (TCP/IP)	X					
Linx Gateway Client Driver	X					
Logix 5550				X		
MicroLogix				X		
Panel View				X		
PLC-5 Channel Zero				X		
SLC500 5/03 Channel Zero				X		
SoftLogix 5 Controller	X					
S&S Technologies 5136-SD/SD2		X				

* 1771-SDNPT driver to support pass through capability from a 1771 based network to DeviceNet.

Please call Rockwell Software Inc. for details or availability of other drivers.

Another view of the same information:

Connectivity	Windows 95/98	Windows NT
Ethernet	✓	✓
1784-KTX / KTXD / PKTX / PCMK / KT / KT2 to DH+	✓	✓
1784-KTX / KTXD / PKTX / PCMK to DH-485	✓	✓
1784-KTC to ControlNet	✓	✓
1784-PCC to ControlNet	✓	✓
DF1 to 1770-KF2 / 1785-KE to DH+	✓	✓
DF1 to 1770-KF3 / 1747-KE / 1747-PIC to DH-485	✓	✓
DF1 to PLC-5/CH0, SLC/CH0, MicroLogix, PanelView	✓	✓
DF1 to 1770-KFC to ControlNet	✓	✓
DF1 to 1770-KFD to DeviceNet	✓	✓
DF1 Polling Master	✓	✓
SS Technologies 5136-SD/SD2 to DH+	✓	✓
SoftLogix 5 Controller	✓	✓
Remote Devices via Linx Gateway	✓	✓

Benefits of RSLinx

- Ease of upgrade to new processors and networks because all of the 32-bit Allen-Bradley drivers are included in one package.
- Compatibility of products from Rockwell Software, Allen-Bradley and third party products or custom solutions using RSLinx's open C API, OPC or AdvanceDDE.
- Synchronous read and write access to processor data in PLC-5 and SLC 500 processors via DDE interface.
- Synchronous and asynchronous read and write access to processor data in PLC-5, PLC-2, PLC-3, PLC-5/250, MicroLogix 1000 and SLC 500 processors via C API.
- Concurrent operation of multiple communication devices.
- Intuitive user interface tested in our Usability Lab.
- Easy Copy/Paste DDE hot link establishment.
- Data sharing with other computers via Network DDE.
- Efficient use of system resources and minimal network traffic via optimized DDE reads.
- Superior speed and reduced network load with block reads and writes.
- Connectivity to legacy networks supported by remote routing through 1785-KA, 1785-KA5, 5130-KA, 5820-EI, 1756-DHRIO, and Remote ControlNet.
- Intuitive system investigation supported by graphical RSWho function and comprehensive diagnostics.
- Assistance is just a click away, via context-sensitive help, "What's This" right click help, and On line books.
- Seamless upgradability from RSLinx Lite or RSLinx OEM.
    - RSLinx Gateway can only be installed from an RSLinx Gateway CD.

RSLinx Lite, RSLinx OEM, RSLinx Professional, RSLinx SDK, RSLinx Gateway

What's the difference?

RSLinx is available in versions to meet the cost and functionality requirements of many different applications. Depending on the version you are running, some functionality may or may not be operational.

To know which version you are running, read RSLinx main title bar text (the line of text at the very top of the main window). If you are running the Lite or OEM version, some of the features documented in the online help file may not be enabled. The following explains the main differences between the different versions of RSLinx:

We've packaged RSLinx to provide the best cost/functionality trade-off to address different market segments:

- ***RSLinx Lite –*** Only used when bundled with other Rockwell Software products that use RSLinx as their communication interface to plant floor devices. No external interfaces are provided for DDE, OPC or custom 'C' applications.

> ***Note****: An RSLinx package that has lost its activation will revert back to RSLinx Lite. Some applications, such as RSView32, can* ***NOT*** *communicate to tags or other PLC data using only RSLinx Lite.*
>
> *Ensure the activation files are safeguarded to protect against loss or damage.*

- ***RSLinx OEM –*** Intended for use with third party HMI's or custom applications. Provides complete driver support for Allen-Bradley devices and networks.

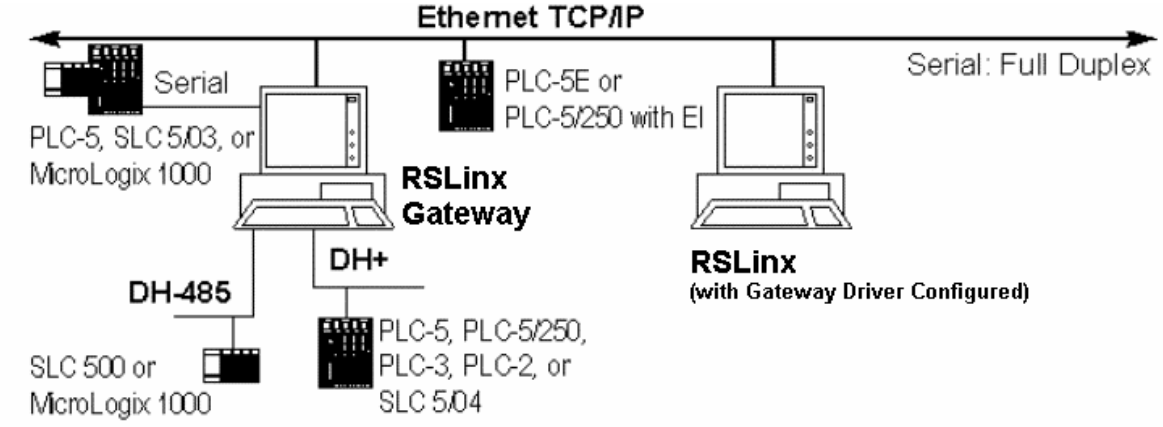
  We expose our AdvanceDDE interface for DDE messaging, OPC COM interface for *local* OPC connectivity and runtime support for custom applications developed with the RSLinx SDK.

- ***RSLinx Professional –*** Full-featured version. Includes all the functionality in RSLinx OEM plus additional features such as: DDE formats (XL_Table, CF_Text) for Microsoft Office or standard DDE client applications, FastDDE for Wonderware clients, and an OPC Automation interface for *local* OPC connectivity.

- ***RSLinx Gateway –*** Includes all the functionality in RSLinx Professional, plus *remote* client connectivity or TCP/IP. Allows additional workstations running RSLinx or an OPC Client application to "gateway" to Allen-Bradley networks connected to the server PC. Can connect from an off-site location using Remote Access Service (RAS).

  With the purchase of RSLinx Gateway, you have a license for up to five concurrent remote connections. More remote connections can be obtained by purchasing additional license packs.

Typical RSLinx Gateway Configuration



- ***RSLinx SDK –*** The Software Development Kit (SDK) provides all the necessary files and documentation for development of custom C/C++ applications to communicate to the RSLinx runtime interface.

  A copy of RSLinx OEM is included with this purchase. End use of the custom application will require additional purchases of RSLinx OEM, RSLinx Professional, or RSLinx Gateway.

Feature	RSLinx OEM	RSLinx	RSLinx Gateway	RSLinx C SDK
Communication Drivers	✔	✔	✔	✔
Driver and Station Diagnostics	✔	✔	✔	✔
AdvanceDDE support	✔	✔	✔	✔
Standard DDE support (CF_Text, XL_Table)		✔	✔	
Local OPC Client connection	✔	✔	✔	✔
Remote OPC Client connection[1]			✔	
Remote Gateway Client connection[1]			✔	
C Software Development Kit				✔
Catalog Numbers	**9355-WABOEM**	**9355-WAB**	**9355-WABGW**	**9355-WABC**

[1]Client license grants 5 concurrent remote connections. Additional client license packs available for purchase

RSLinx Requirements

Software requirements

RSLinx requires one of the following software environments:

- Microsoft Windows NT Version 4.0 (Service Pack 3 or later recommended). Because RSLinx takes advantage of features not available in Windows NT prior to Version 4.0, RSLinx is only supported on Windows NT Version 4.0 or later.
- Microsoft Windows 95 with DCOM for Windows 95 installed. DCOM for Windows 95 must be installed before attempting to install RSLinx, or the RSLinx installation will fail.
  - DCOM for Windows 95 can be installed from the RSLinx CD or downloaded Microsoft's DCOM95 website (www.microsoft.com/com/dcom.asp).
- Microsoft Windows 98.

***Note**: RSLinx 2.00.97.30 will not install on Windows NT 4.0 with Service Pack 4. The version of Harmony that is installed with RSLinx 2.00.97.30 will not install on Windows NT 4.0 with Service Pack 4. The installation hangs at about 50% (during the Harmony Installation).*

The solution is to download the file "HarmonyRT.zip" from Support Library Technote ID: P1040.

This is not a problem with the 2.1 version of RSLinx.

Hardware requirements

RSLinx has the following minimum hardware requirements:

- An Intel 486/66 or Pentium processor with at least 16 megabytes of RAM. This version of RSLinx will not run on Alpha, MIPS, or Power PC processors. The versions of Windows NT for different processors are not binary-compatible.
- At least 15 megabytes of available hard disk space. 20 megabytes of available hard disk space may be required for specific application features.
- A 16-color, SVGA display with 800 by 600 or greater resolution.
- A Windows-compatible pointing device.

RSLinx Help Systems

Rockwell Software Technical Support

Most questions can be answered with RSLinx's many internal help options without a call to Rockwell Software's Technical Support, but if you must call, expect superior customer service.

Rockwell Software is the first company in the automation industry to receive the Support Center Practices (SCP) Certification from ZD Service and Support Consultants for exemplary customer service in the area of technical support and training programs.

Rockwell Software plans to continuously improve in the areas of technical software support and customer service to assure that their customers receive world class assistance.

The Software Support Professionals Association (SSPA) coveted Software Technical Assistance Recognition (STAR) award for Most Improved was another recent outstanding achievement!

Just call Rockwell Software support at (440) 646-7800 (Mayfield Village, Ohio), Monday through Friday, 8:00 A.M. to 5:00 P.M. Eastern Time.

You can also fax your question to us at (440) 646-7801.

RSLinx Internal Help

Help with RSLinx can be easily accessed by either pressing the F1 key or using the Help Menu Options seen below. It uses a standard help format you may have seen in other products.

Enhancements to the help system include:

- What's This? help in most areas of RSLinx by right-clicking an option. Press the What's This? button and a help topic popup will appear.
- Similar to What's This?, the Context Help icon has been added to the toolbar. Just click the ? icon  and the cursor will change to this: 

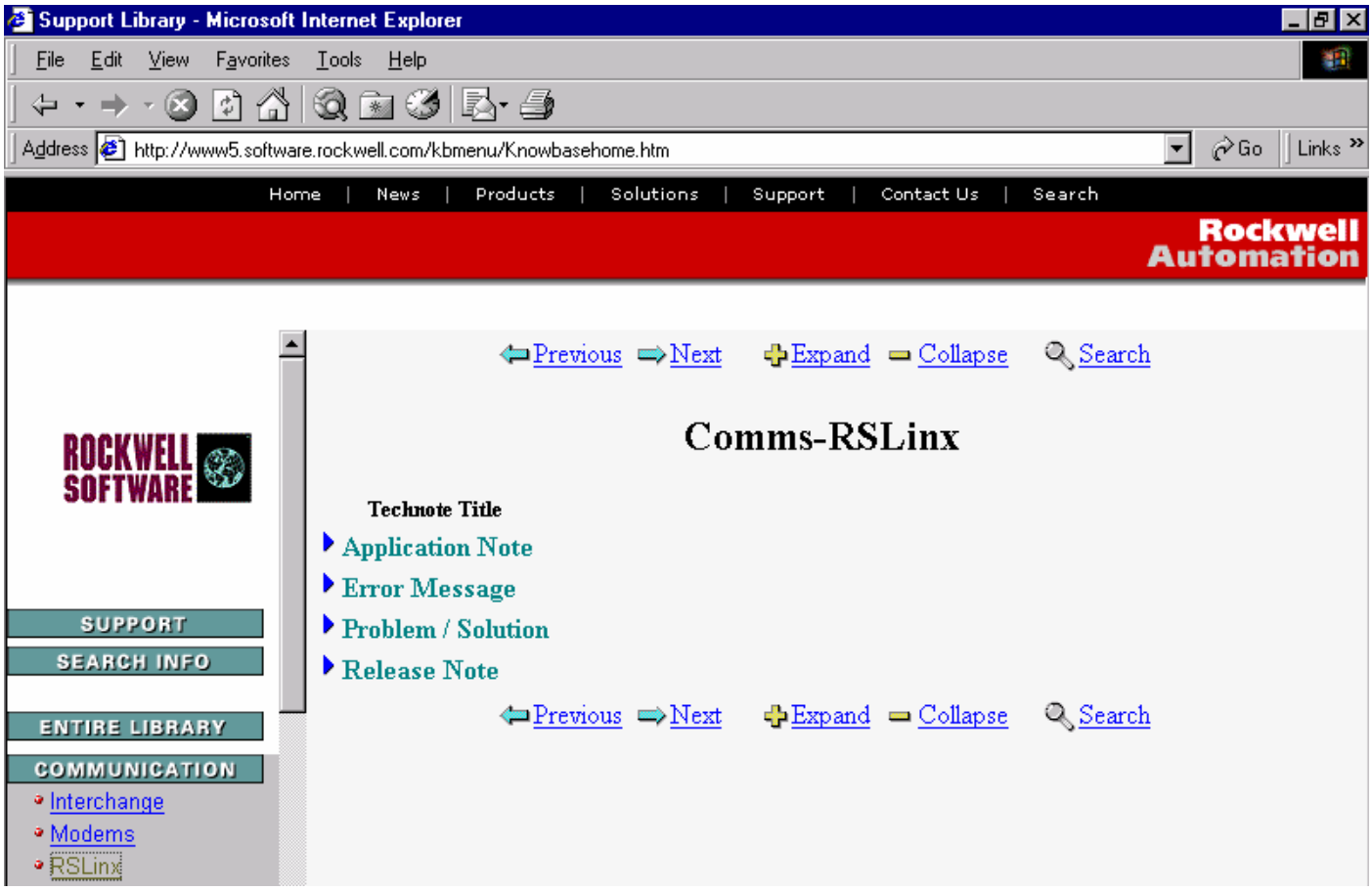
  Click the topic and the same help topic popup will appear.

- Online books that include a Hardware Configuration Manual and the RSLinx Getting Started Guide in Adobe Acrobat .pdf format. The reader tool is included and you may print the file if you'd like.
- RSI on the Web takes you to the main Rockwell Software website.
- Copy Protection help
- Quick Start – walks you through the four steps of creating a DDE hot link.

Help on the Web

On the Rockwell Software website, support area:
http://www.software.rockwell.com/support/

Support Library
Browse our on-line Support Library for tips and answers to your most common problems. Our support technicians enter tech notes in the Support Library based on calls to our Support group.



Reset Codes
Have you lost a Rockwell Software product's activation due to some type of hard drive damage or corruption? A Reset Code can help you get up and running quickly. This requires that you have your MASTER disk available.

Support Request
Submit a Support Request using your Web browser, and we will either contact you by telephone or e-mail as quickly as possible. Have your product serial number available to complete the process.

Software Updates
If your software is currently under warranty or a support contract, you may be able to download a Software Update.

Training
Our Training Catalog is available on-line, including detailed course descriptions, schedules and enrollment information. You'll find a range of courses covering Rockwell Software and Microsoft technologies. You can even sign up for a course on-line!

Support Guide
The purpose of our Support Guide is to describe our support services for you. Our most important goal is that you are completely satisfied and comfortable using Rockwell Software products. That's where our technical expertise and variety of support services can really make a difference.

Downloads
Search our database of Downloadable Files for Demos, General Downloads, Presentations, Promotional Information, and Customer Feedback! (This does not include On-line Software Updates.)

Transfer Registration
Transfer your Product Registration using your Web browser! If we need additional information to complete the transfer, we will contact you by E-mail.

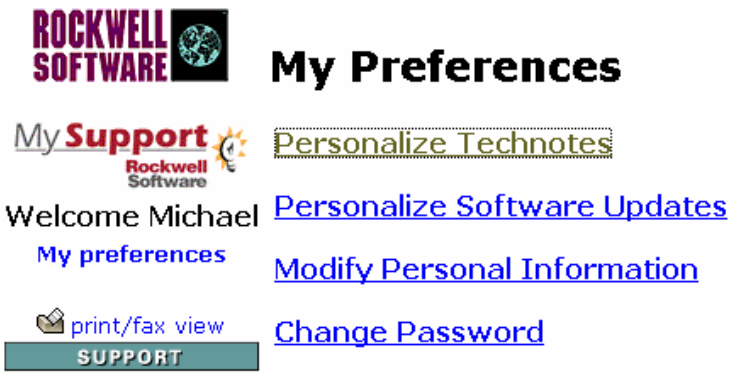
The Software Connection Magazine
- Your Link to Software Support and Product News

If you have not seen the latest issue in print, you can find it on-line, as well as past issues dating back to 1995.

Check out the URL: http://www.software.rockwell.com/corporate/swc

My Support

A new support feature on our website allows you to personalize technotes and personalize software updates.



You can select the products of interest to you and can specify your preferred delivery option(s).

Application Notes and Suggestions

- With a ControlLogix 5550, unsolicited messages work on only an Ethernet network or a DH+ network.
- To receive unsolicited DDE messages from a processor through a DHRIO module, and through an ENET module over an Ethernet network, select your DDE topic from the DDE/OPC Topic Configuration dialog box. Then, select Advanced Communication, click Remote, and manually enter the Remote Link ID of the DHRIO channel through which the message is being sent.
- Newly created DDE topics default to 20 for Limit Maximum Packets. This makes the updates more symmetrical for the group of optimized packets for each topic. For the same functionality on older DDE topics, set Limit Maximum Packets on the DDE/OPC Topic Configuration dialog box, Data Collection tab to 20, or the best value for the PLC and access method.
- When configuring DDE/OPC topics to receive unsolicited messages from a ControlNet device, from the DDE/OPC Topic Configuration dialog box, select the Advanced Communication tab and select Local route for Local or Remote Addressing.
- To ensure that unsolicited messages coming to RSLinx work properly when setting up a DDE topic to a remote processor via ControlLogix Gateway, set the Remote Link ID to match the Link ID of the remote module (such as DHRIO).
- Block reads and writes do not work with user defined types - (ControlLogix 5550).
- In Windows NT, the PIC/AIC+ and the DF1 Polling Master / Slave drivers (with RTS control enabled only) drivers need their own serial driver (RSSERIAL.SYS rather than SERIAL.SYS) and cannot therefore share an interrupt with another serial device.
- When using auto configuration on a 1770-KF2/KF3 device, you must set the Station Number to that of the KF2/KF3 before clicking Auto-Configure. Otherwise, Auto-Configure tries to detect the device at the currently selected station, and if the device is a PLC, Auto-Configure configures to the PLC rather than the KF2/KF3.
- If you restore a Windows NT configuration, and you have a configured KTC(X) or PCMK driver, you must reboot the computer to start the driver.
- After the first time you configure a PCC driver in Windows NT, you must reboot your computer before using the driver.
- When you install RSLinx, DTL32.DLL remains in the system directory if it is locked by another application.
- You must have Windows NT Administrative privileges to install RSLinx and/or configure RSLinx drivers.

Application Notes and Suggestions (continued)

- When receiving unsolicited messages from an SLC 500 processor via a 1785-KA5 Bridge, the DDE topic's remote configuration must be set to Internet mode.
  - Also, in the SLC 500 message instruction, the remote bridge node address must be set to 0.
  - Also, unsolicited messages sent from a PLC or other node to RSLinx will cause errors if no DDE clients or C-API applications are running to receive them.
- When downloading a program to an SLC 500, stop all data connections active to that station. Under certain instances, the processor may fault if applications continue to poll the SLC during a download.
- If you are running Windows 95 using the 1747-PIC driver with WINtelligent LINX, but are no longer using WINtelligent LINX, we recommend you change the following entry in your system.ini file (located in \Windows directory). Under the [boot] section of the system.ini file, change **comm.drv = icomcomm.drv** to **comm.drv = comm.drv** After changing this entry, restart Windows 95.
- RSLinx does not automatically create a NetDDE share entry in the NT or Windows 95 registry for configured topics. For help on enabling NetDDE and creating NetDDE shares, refer to the NetDDE help page.
- RSLinx will not reread a User Defined Structure list for SoftLogix until the topic is stopped and restarted.
- RSLinx supports no-interrupt configurations when using Allen-Bradley network cards; however, we recommend assigning an interrupt to increase driver performance.
- If you have Auto Hide on for the Windows task bar, every time you go to the task bar (if animation is on), it reduces your packets/sec speed due to computation times.
- Do not use the power saver option on your computer when running RSLinx. Use either the BIOS or Control Panel to disable this option.

Application Notes and Suggestions (continued)

Downgrading to a previous version of RSLinx

If you have RSLinx Version 2.1 installed, but wish to downgrade to a previous version of RSLinx, complete the following:

1. Select **DDE > Topic Configuration**, and delete any DDE topics configured in RSLinx 2.1.
2. From the Start menu, select **Programs > Rockwell Software > RSLinx > Uninstall RSLinx 2.1** (assuming you used the defaults when you installed RSLinx 2.1). You may choose freely to preserve or delete configurations.

   You may also choose to uninstall RSLinx from the Control Panel's Add/Remove Programs selection.

3. Reboot your computer.
4. Install the previous version of RSLinx.

To use with A.I. Series Ladder Logistics

In Windows NT 4.0, this version of RSLinx is only compatible with versions:

- 7.21 or higher of PLC-5 A.I. Series Ladder Logistics
- 8.14 or higher of PLC-500 A.I. Series Ladder Logistics
- 6.19 or higher of PLC-3 A.I. Series Ladder Logistics
- 1.39 or higher of PLC-5/250 A.I. Series Ladder Logistics

In Windows 95, this version of RSLinx is only compatible with versions:

- 8.00 or higher of PLC-5 A.I. Series Ladder Logistics

Windows 95 support for other A.I. and RSLinx will be added in future revisions of A.I. programming software. If you are using an older version and are under support, call customer support and request an upgrade.

Application Notes and Suggestions (continued)

1784-PCMK and PCMCIA information and Windows NT limitations

When using a 1784-PCMK (PCMCIA or “PC” card) communications card, Windows NT has the following limitations:

- When a 1784-PCMK is reconfigured through RSLinx, your Windows NT system must be rebooted for the changes to take effect.
- Only one PCMCIA card of the same type may be used at one time.
  This means that you can only use one 1784-PCMK at a time or one PCMCIA Ethernet card at a time with RSLinx. You can also use one 1784-PCMK and one PCMCIA Ethernet card at the same time
- Windows NT and RSLinx do not support dynamic removal or insertion of PCMCIA cards. If you remove or insert a PCMK card with RSLinx running, you must re-start your Windows NT system for the card to be recognized.
- It is easier to configure the PCMK card when using board addresses at or above D000.

Running RSLinx on NT with computers that are not networked

When running RSLinx on NT with the 1784-KT/KTX(D)/KTC/PCMK, RSLinx requires NDIS services to be running.

Windows NT does not install or start NDIS services unless it has been configured to use a network driver. To make sure that NDIS services are installed and started, a network protocol must be selected in Windows NT.

If you attempt to configure an RSLinx driver, and network services are not installed on your workstation, you will see the following succession of message dialogs. Click OK to proceed through each.

- `System Process - System Error\Object Name not found.`
- `RSLinx\StartServiceError: Unable to start NDIS Kernel Mode Mac Driver. See Application Event Log for Specific error(s).`
- `System Process - System Error\Object Name not found.`
- `RSLinx\StartServiceError: Unable to start NDIS Kernel Mode Mac Driver. See Application Event Log for Specific error(s).`
- `RSLinx\StartServiceError: Unable to start NDIS Kernel Mode Mac Driver. See Application Event Log for Specific error(s).`

RSLinx will be running but drivers will not have been started. If these messages occur when you attempt to configure an RSLinx driver, you must install network services on your workstation.

Application Notes and Suggestions (continued)

Running RSLinx on machines not on a network

> ***Important:*** *This procedure requires that you log into Windows NT with system administrator privileges. For RSLinx to work properly in Windows NT 4.0, you must have installed RSLinx Release 1.50.58 or later.*
>
> *If you have an earlier version of RSLinx, please upgrade to the latest version of RSLinx.*

1. Start Windows NT 4.0 and go into the **Control Panel**.
2. Double-click the **Network** icon. If NT asks if you want to install Windows NT Networking, click **Yes**.
3. NT takes you through the Network Setup Wizard. When NT asks how you are connected to the Network, choose **Wired to the Network**.
4. Do not have NT search for a adapter. Select **MS Loopback Adapter**.
5. Choose TCP/IP Protocol.
6. Click **Next** to accept the four other services NT will load.
7. Place your NT 4.0 CD in the drive and click **Continue**.
8. During the install, you will be prompted for a Frame Type. Choose **802.3**.
9. When asked if you want to use DHCP to configure your TCP/IP, click **No**.
10. When prompted to configure your TCP/IP Properties, use an IP Address of **100.100.100.100** and a Subnet Mask of **255.255.255.0** and click **OK**.
11. Choose **Workgroup** and insert a name in the box.

You must restart your system for these changes to take effect. When Windows NT restarts, you may go into RSLinx and configure your drivers.

Application Notes and Suggestions (continued)

Routing through bridge devices

- RSLinx can not communicate through an external bridge device when using the 1756-DHRIO bridge device with the TCP or CNET driver.
  The problem exists when the first bridge is a 1756-DHRIO device and the second bridge is not a 1756-DHRIO device (for example, Pyramid Integrator). In this case, the source link ID remains at zero and the response packets fail. The following limitations currently exist:
- Offlink routing works through one bridge.
- Offlink routing works through two or more bridges if the first bridge is a 1756-DHRIO module.
- Offlink routing does not work through two or more bridges if the first bridge is something other than 1756-DHRIO device.
- You cannot go from a DHRIO module through a 1785-KA bridge device to a DH network.
- To go from a DHRIO module through a 1785-KA5 bridge device to a DH485 network, the DHRIO module must be configured using the ControlLogix Gateway Configuration tool. For example, if a 1785-KA5 bridge is on a DH+ network at node 4 and the DH485 LinkID is 13, complete the following:
  1. Select **File > Browse Network**.
  2. Select the DHRIO module.
  3. Go to the routing table configuration tab.
  4. Right-click on the channel being used and select **Add Module**.
  5. Select the DH+ Bridge.
  6. Enter the DH+ node number of the KA5 (4 in this example), and the LinkID of the DH485 (13 in this example).
  7. Click **Apply**. You can now browse through the KA5 module.

Training Section 2: Installing RSLinx

RSLinx is installed using a standard Windows setup program.
Your RSLinx package may consist either of a CD or of several disks (not both).
If disks, you can start with the disk labeled *Disk 1.*

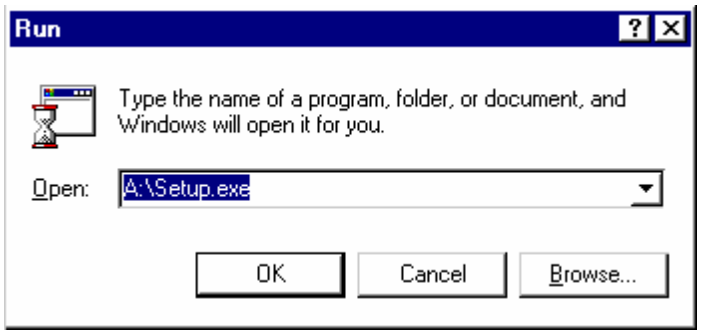
Complete the following to install RSLinx:

> ***Note**: In Microsoft Windows NT, as with any other network software, you must have administrator privileges to install RSLinx software.*

1. Insert RSLinx diskette 1 into a floppy diskette drive (or for CD installation, insert the CD in the CD drive).

2. Click the Start button on the taskbar, then Run, and Browse to the drive that contains the RSLinx diskette or CD.

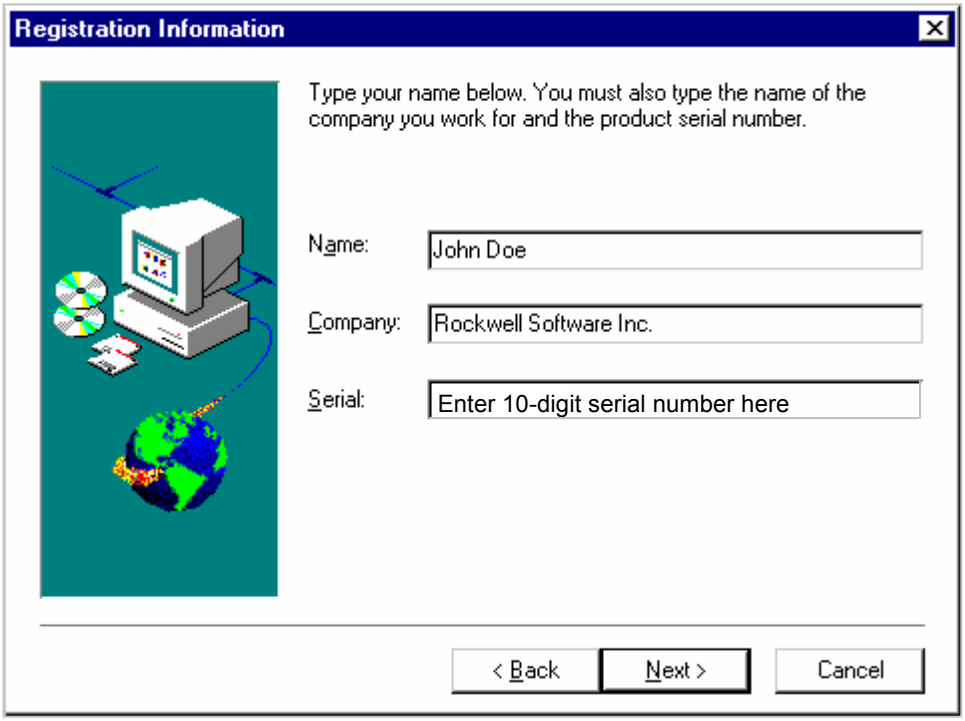
3. Find and double-click SETUP.EXE.



4. Click OK.

5. Follow the directions that appear on the screen.
   - One of the final steps in the installation process requires the activation to be installed. It is the "key" to activating the software and without it, RSLinx is only RSLinx Lite.
     - The Lite version has no DDE or OPC capability.

6. A new installation step will require a 10-digit number.
   - The setup program will not all you to proceed until after the serial number found on the RSLinx activation disk is entered.

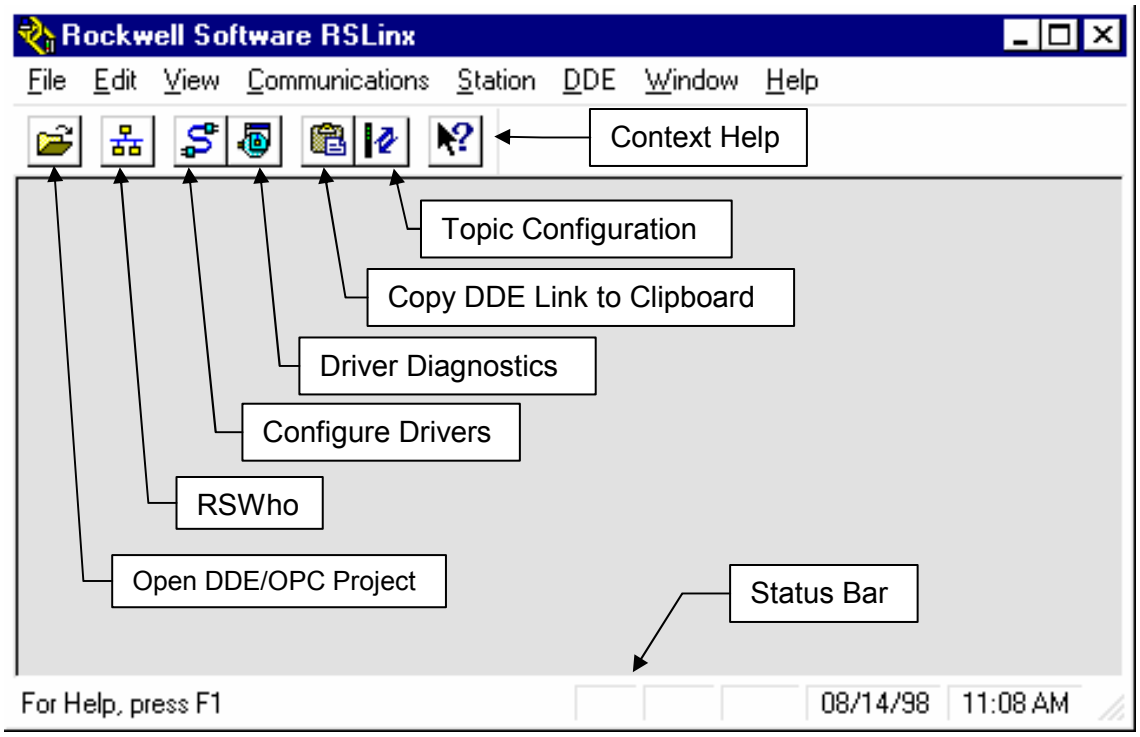


7. Click Next and proceed with the installation.

8. After completing the RSLinx installation, remove the RSLinx diskettes and store them in a safe place, especially the master activation disk.
   - Even an activation disk with NO activations left on it can help you recover from a lost or damaged activation. See the Help Menu / Copy Protection section for instructions on how to use the key disk feature.

Training Section 3: Configuration of Communication Hardware, RSWho and Diagnostics

1. Open RSLinx. The following screen will appear:

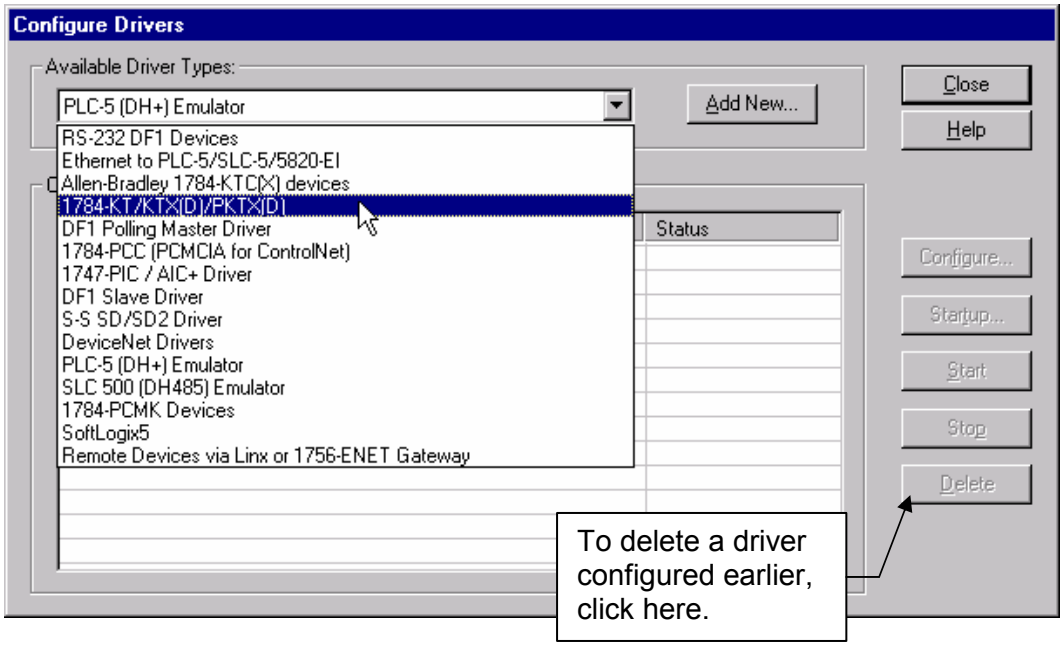


The first time you start-up a newly installed RSLinx package you are required to configure communications hardware.

In this section we will go through an example of how to configure communication hardware and how to use the new RSWho (Who Active) and highway diagnostics.

Lab / Discussion:

1. Start-up RSLinx.
2. Click the Configure Communications Hardware icon.  The Configure Drivers box, seen below, will appear.

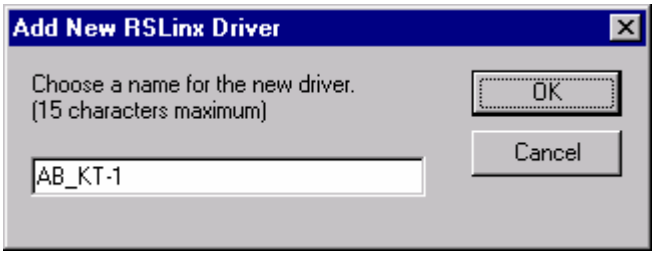


3. Pick the hardware device from the Available Drivers list that you intend to use to communicate with the PLC world and press Add New…

    - For the example in this training exercise, we will select the 1784 KT/KTX(D)/PCMK/PKTX(D).

***Note**: If you are using a hardware device other than the KT that is used in this example, you can choose that alternate device.*

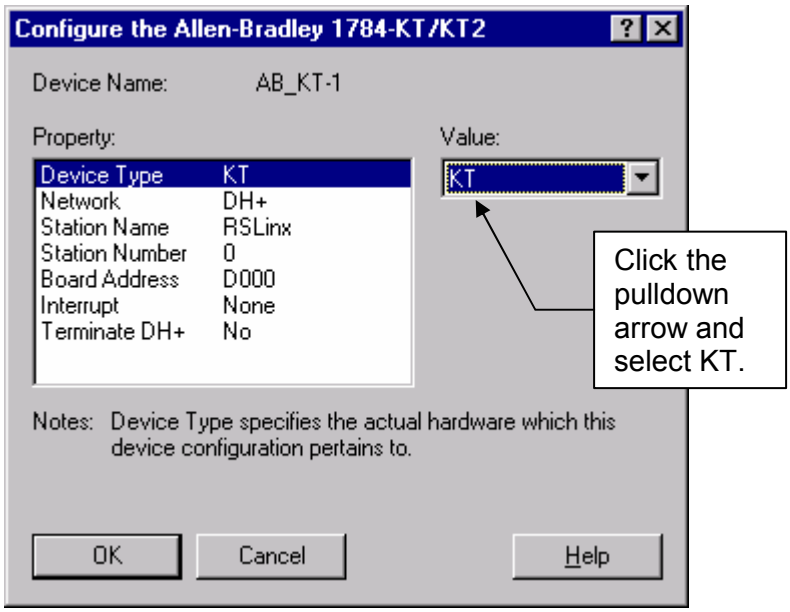
4. Click Add New. The following dialog will appear to name the new driver:



- Use the default AB_KT-1

5. Click OK

- Then the following dialog will appear to choose the type of KT card and begin configuration:



***Tip**: Rockwell Software provides a utility to read your WINtelligent LINX configuration and display it for use in configuring RSLinx.*

This utility is available on the Rockwell Software website (www.software.rockwell.com) or by contacting Technical Support at (440) 646-7800.

Configuring the 1784 KT

1. To configure the KT, take the following steps (each item is selected and configured by clicking on that item's row in the previous box).

    a) Select the hardware (Device Type) by. Choose KT.

    b) Select the Network Type (DH+ or DH485) by clicking the Network Type item and then clicking the pull down arrow on the right side of this box to select the correct option. Choose DH+.

    c) Create a Station Name up to eight characters. In this case, use the default: RSLinx.

    d) Enter the Station Number - a valid, non-duplicate station address for the DH+. (Range: 0-77 Octal)

    e) Pick the correct board hexadecimal address for the KT.
    This must match the settings made on the KT DIP switches.
        - If you are unsure of your KT board's dip switch settings, you will need to check the board and the installation documentation to match the dip switch settings to their address.

    f) Select the correct interrupt number. This should correspond to that set by the hardware jumper on the KT board.
        - An interrupt (or IRQ) is not required, but for best performance, select an unused interrupt of either 3, 4, 5 or 7.

    g) If this is the last station on the network, it is important to select YES to terminate the DH+.
        - Turning on terminate DH+ will switch in a terminating resistor, as a PLC would have if it was the last station on the network.

    h) Click OK.
        - RSLinx will now find and configure the KT card.

Take note of the new AB_KT-1 driver that is displayed in the Configured Drivers section of the Configure Hardware dialog box.

For the first KT card configured this will be AB_KT-1. The full configuration should read:

AB_KT-1 DH+ Sta:0 Addr:D800 RUNNING

RSLinx supports multiple communication hardware devices, so the next KT configured will be named AB_KT-2 (by default, although you can change it)
and so on.

The only limits to multiple communication drivers are the physical locations for devices in your PC, memory addresses, and Windows resources.

RSWho

RSWho is RSLinx's network browser interface. RSWho allows you to view all the active network connections from a single screen.

The left pane of RSWho is the Tree Control, which shows networks and devices. The right pane is the List Control, which shows all members of a collection.
A collection is a network, or a device that is a bridge.

Features:

- RSWho browsing is restricted to one network at a time.
- RSWho list view has two views, large icon and detail.
- RSWho supports ascending and descending sorting for columns in list view.
- RSWho supports persistence of window settings including column width and splitter position.
- RSWho is started automatically and maximized when RSLinx starts up.

***Important**: A device that appears with a red X over it indicates that RSWho previously recognized this device, but now it can not. The red X indicates a communication status error, such as unplugging a recognized device.*

***Tip**: To remove multiple "dead nodes" delete a driver from RSWho, then collapse and expand its parent branch.*

Browsing

The RSWho icon indicates a network. If the icon is animated, the network is being browsed. Click on a network or device to start browsing.

When the Autobrowse checkbox is selected, RSWho continuously browses the selected network. If Autobrowse is cleared, the Refresh button is active. Clicking Refresh instructs RSWho to perform one browse cycle of the selected device. Since Refresh only performs one browse cycle, clicking Refresh multiple times may be necessary to discover everything on the network.

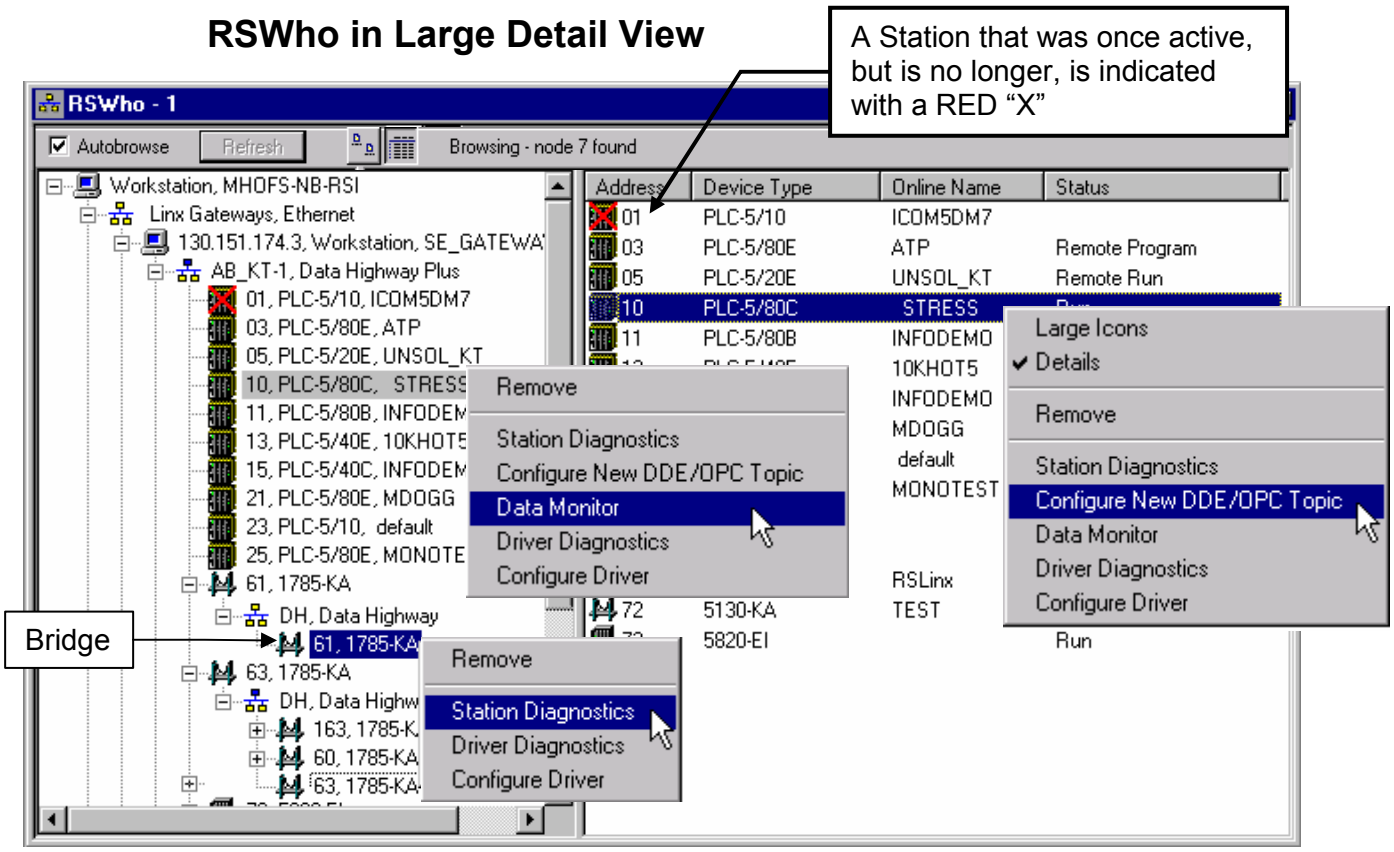
When the network or device is collapsed (indicated by the + sign), click + or double-click the network or device icon next to the + to expand the view and begin browsing. When the network or device is expanded (indicated by the – sign), click – or double-click the network or device icon next to the – to collapse the view.

Right-click on a supported device to select Station Diagnostics or other supported services for that device.

Lab / Discussion

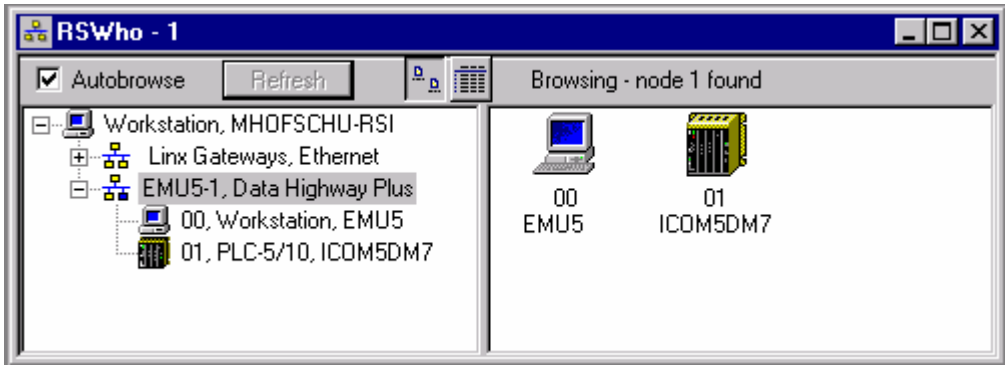
1. Check if the connection to the DH+ is good by opening an RSWho.
   - Click the RSWho icon.
     - Alternatively, open an RSWho by selecting *Communications / RSWho* from the top menu of RSLinx.
2. A right-click on an RSWho screen will allow you select many options, depending on where you right-click, the RSWho below shows many of the RSWho right-click menus.
   - Note: you can not make all right-click menus appear at the same time

RSWho in Large Detail View



***Note**: RSWho will not browse through SoftLogix, Ethernet, or a Linx Gateway on a remote RSLinx Gateway*

RSWho in Large Icon View



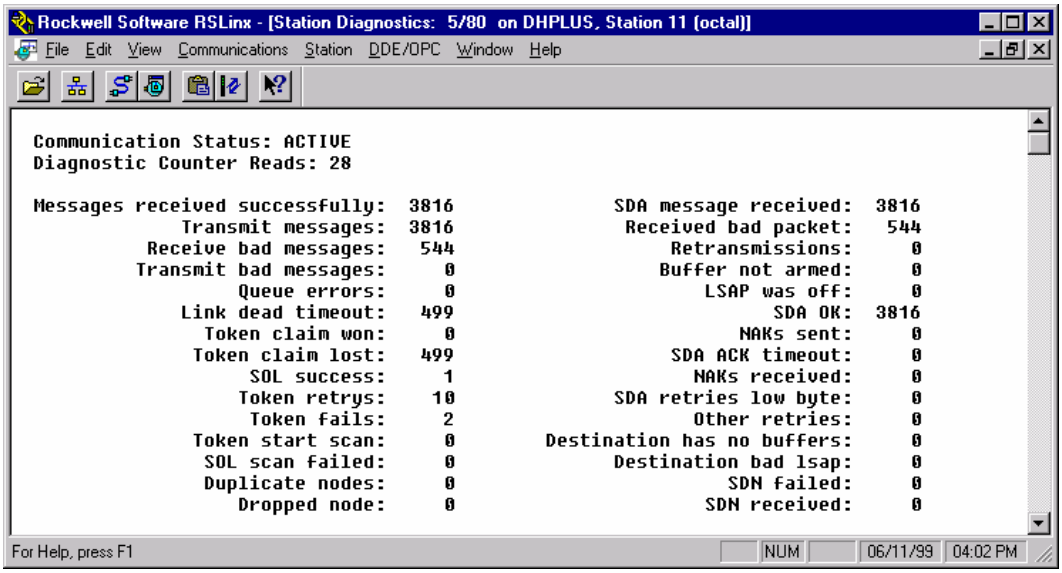
***Note**: It is <u>not</u> a good idea to let RSWho screens run all the time in Autobrowse mode. To keep an RSWho screen fresh, RSLinx must continually poll all the processors on the network selected.*

This generates a lot of network traffic and increases the load on RSLinx's network drivers. Rockwell Software suggests that an RSWho only be used when necessary, then closed.

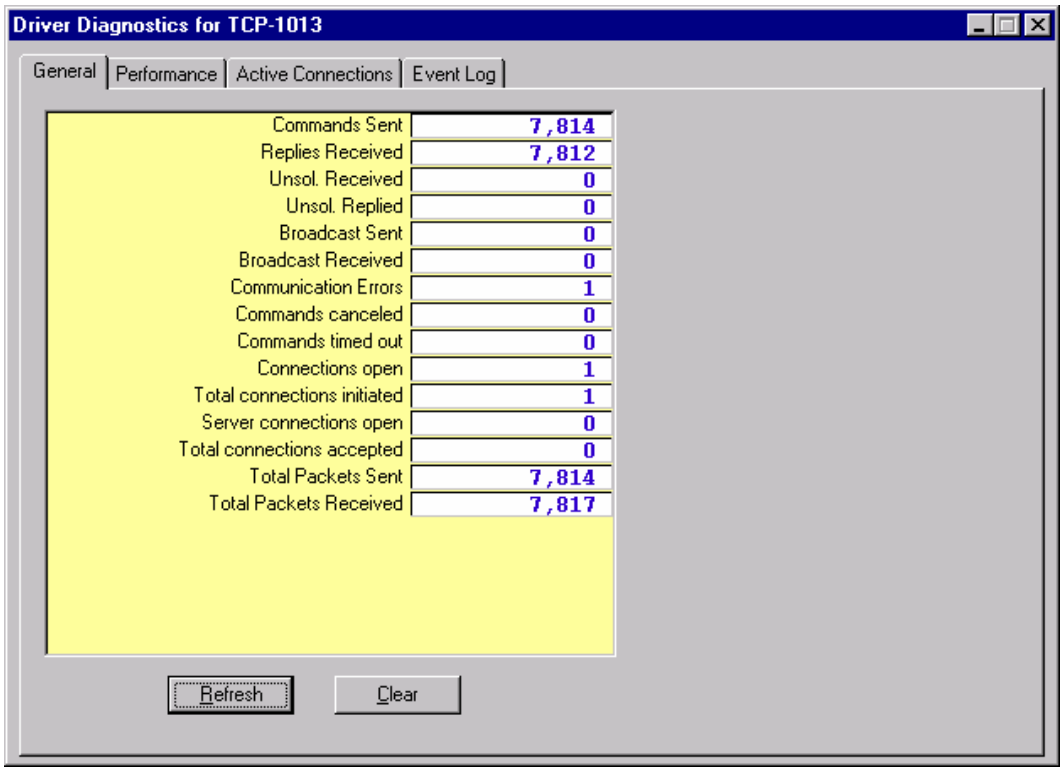
As an alternative if you would like to keep an RSWho open, uncheck the refresh button and if you think the network might have changed, or you encounter a problem, just press the manual Refresh button.

3. Note that each device on the DH+ is identified by its own icon. Also displayed are the station or project name and the station number.

4. In either the right or left pane of the RSWho, right-click on one of the PLC stations and select Station Diagnostics. The following will appear:



5. Right-click on the KT in the RSWho window. Select Driver Diagnostics.
   - When you are finished looking at the diagnostics, close the RSWho window and diagnostics windows by clicking the “Xs” in the upper right of the window.

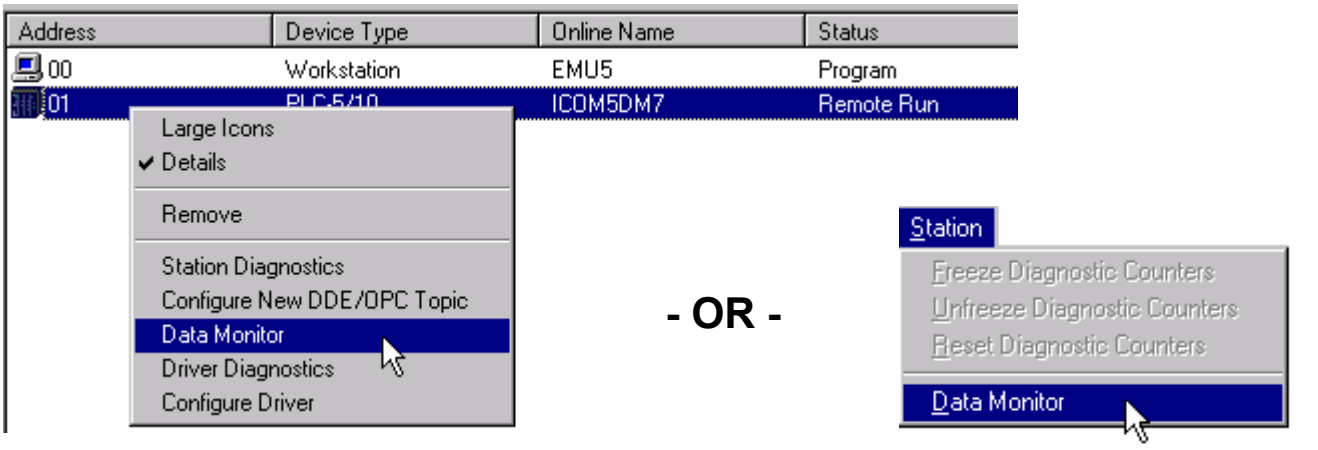


Training Section 4: Using the Data Table Monitor

Data Table Monitors

The Data Monitor can now view the activity in your data table files for both PLC-5 data as well as SLC data. This procedure is called *data monitoring*.

The Data Monitor window is modeless; therefore, they can remain on the desktop if you minimize RSLinx



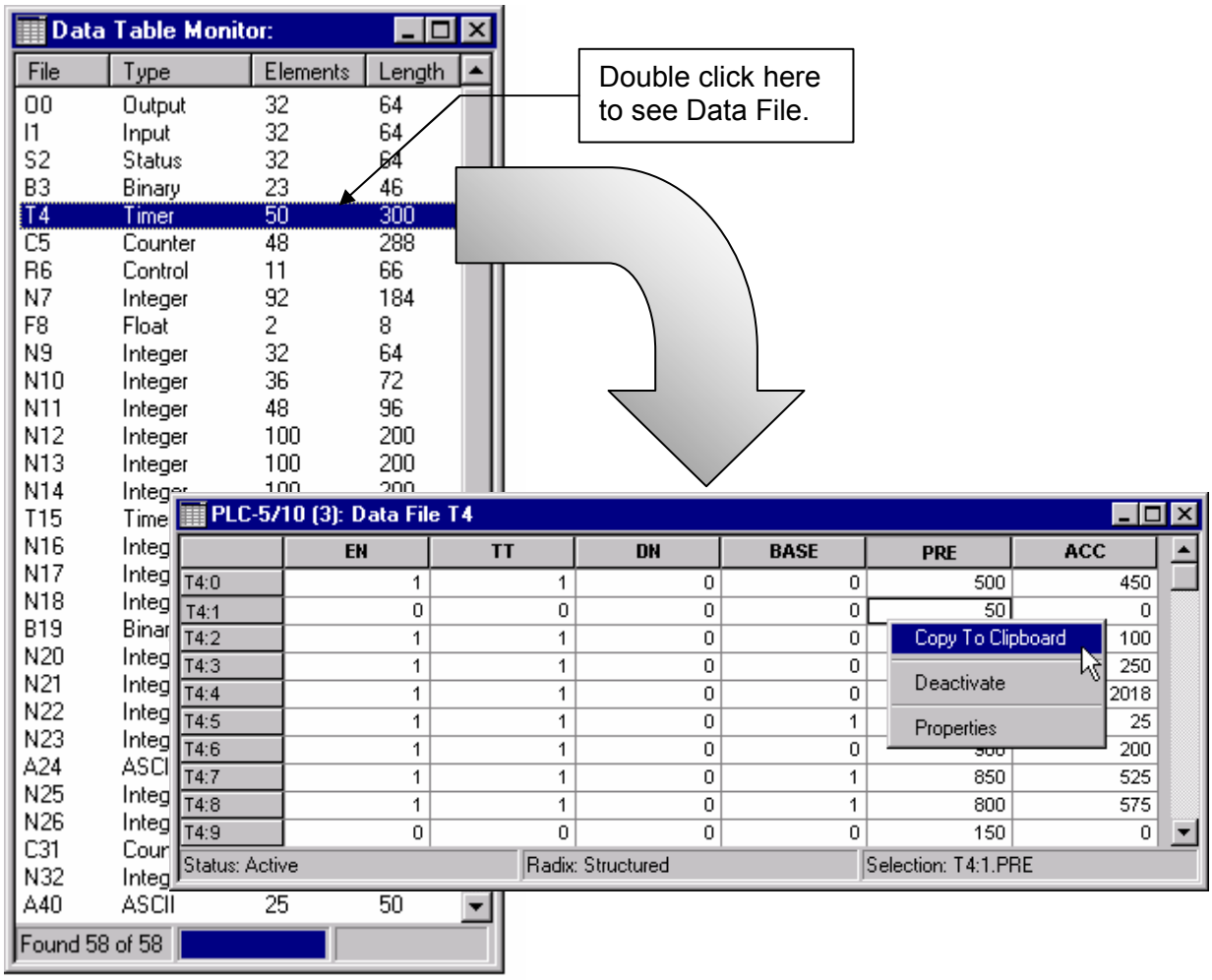
When you are monitoring your data, the information displays in either a data file list window or a data table window. Data monitoring allows you to:

- Select data files to be viewed
- View the data files in a grid format
- View more than one data file at the same time

The RSLinx data table monitor is not available with RSLinx Lite or RSLinx OEM.

> ***Important**: The data table monitor is read-only. Values in the data files cannot be modified directly from the data monitor window. However, elements can be copied to the clipboard and read or written with DDE. Input and Output files are protected against writes with DDE.*

The following fields display on the Data Monitor dialog box:

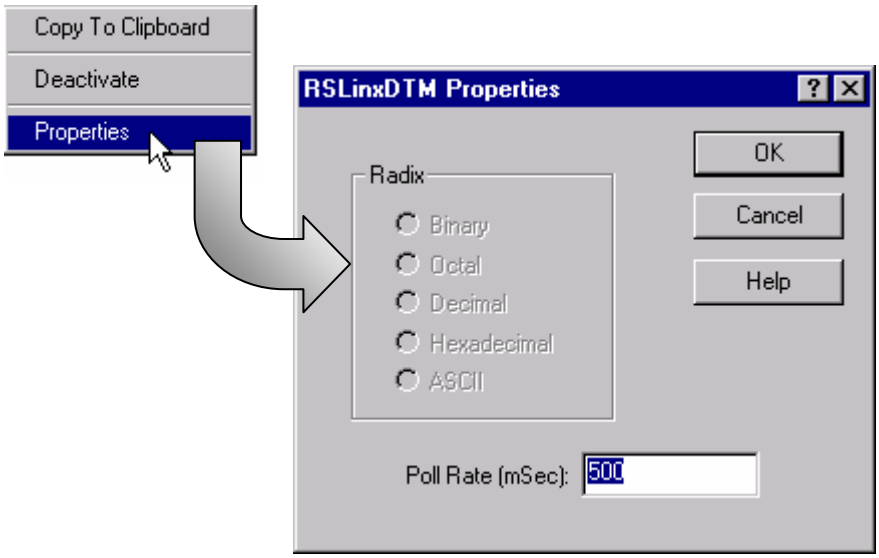


Data Grid displays data values for addresses in the data table file (even if they are not currently being used). The width of the grid can be adjusted to meet your display requirements. When you click a position in this grid, the address displays in the field directly below.

Offset displays the file type and number and the data file element.

Columns - Click the arrow to the right of this drop-down list box to change the number of elements that can be shown in a row horizontally across the data grid. If you view Binary data the default is 16 and cannot be modified.

The Data Table's Properties option allows you to change the poll rate and radix of the data (Binary, Octal, Decimal, Hexadecimal, or ASCII.)



Training Section 5: DDE and How It Works

Dynamic Data Exchange (DDE) is an interprocess communications specification that is built into the Microsoft Windows operating system.

> ***Note**: AdvanceDDE protocol is the only version supported by RSLinx-OEM. DDE is not available with RSLinx Lite.*

It allows Windows programs that support DDE to exchange data between themselves. All communications occur within the Windows environment, essentially transparent to the users, using very simple communication instructions.

> ***Note**: Not all applications that run under Microsoft Windows support DDE. Check with an application's manufacturer before purchasing an application for use with RSLinx.*

The easiest way to visualize Dynamic Data Exchange is to think of a conversation between people in a room. These people represent the different Microsoft Windows applications (software programs) running on your computer.

Some of the people in the room ask questions (clients), others provide answers (servers), and some both ask questions and provide answers (client/servers).

- In other words, the client initiates or requests a data transfer and the servers respond to the request by providing or accepting data.

Dynamic Data Exchange questions are quite generic. When Microsoft Excel asks RSLinx a question, it is merely looking for data. Excel does not know where the data is coming from. Excel only knows that it asked RSLinx a question, and if RSLinx has an answer, then the data will be provided.

DDE Continued - *Application, Topic, Item*

Each Dynamic Data Exchange question consists of three parts:

- To **whom** am I talking (i.e., which *APPLICATION* gets the question)?
- What **subject** are we talking about (i.e., what is the general *TOPIC*)?
- Which **data** do I require (i.e., what specific *ITEM* are we talking about)?

In other words, **who is talking? what's the general topic?**
what specific data?

Windows DDE uses the words: ***Application, Topic, Item*** *to* describe the DDE conversation that takes place between programs running under Windows (e.g., RSLinx and Excel).

Definitions of *Application, Topic,* and *Item*:

Application: When you use a Microsoft Windows application to obtain data from another Windows application, you must provide the name of the *Application* you wish to respond to your question. The *Application* refers to the program you are running, such as RSLinx would be the application that some client (such as Excel) would talk to (it usually refers to the executable program file, e.g., *RSLinx.exe*).

To use the previous example, when you pose a question to your group of people, you must tell them **whom** you want a response from. When you ask RSLinx for data from a PLC on a highway, the *Application* name to use is **RSLinx**. (**RSLinx.exe** is the name of the RSLinx executable file and thus the name to use for DDE communications no matter what you have named the RSLinx icon).

Topic: Once the *Application* is known, we must determine which *Topic* to discuss. To use the previous example, when you pose a question to an individual, you must select a general **Topic** to talk about. With DDE, available *Topics* are determined by the *Application.* The person asking the question must choose an available topic, or data exchange cannot take place.

In RSLinx, a *Topic* will consist of the information necessary to get us communicating with a particular PLC.

A DDE *Project* is a storage container for one or more DDE Topics, as a folder holds files in Windows. By grouping topics together in a project, you can make multiple topics available at the same time.

Item: After the *Application* and *Topic* are known, your Microsoft Windows application must provide RSLinx with the specific *Item* to discuss. To use the example of the conversing people: if the individual (**who**) and general **topic** of conversation are known, you must determine the exact **data** you require.

In the case of Excel, we might know we want to talk to RSLinx (Application) about a PLC-5 at a certain station (Topic) (that is, we want to send data from a PLC-5 to an Excel spreadsheet).

The last part of the question, *Item,* determines which specific data table address to return to your Windows application.

Example:	**Application**	**Topic**	**Item**
	RSLinx	PLC5Topic1	C5:0.acc
	(RSLinx)	(specific PLC)	(specific counter value)

Where RSLinx is the *application* name, PLC5Topic1 is an example *topic* name, and C5:0.acc is an example *item*, in this case a counter accumulator in a PLC-5.

The **Application, Topic**, and **Item** are basic parts of all DDE conversations. Each DDE compatible Windows program has its own unique syntax. The syntax are special markings, such as ‘ = | and ! . Consult the documentation of your Windows program to determine the exact procedure and syntax used for DDE conversations.

Data Format: AdvanceDDE, XL_Table and CF_Text
A brief discussion of available Data Formats for DDE communications follows.
It is intended to provide you with a general understanding of Data Format from a user standpoint.

Normally no user intervention or control of data formats is possible, and for the most part, the use of different data formats is transparent to the user. There is one major exception - you will certainly notice a vast improvement of the speed at which your data is transferred if one of the "higher" levels of the available data formats is used.

Microsoft has implemented two Data Formats that are available for DDE conversations: **CF_Text** and **XL_Table**.

> CF_Text is the simplest and easiest to implement. All DDE compliant applications should be able to understand this data format. It is the basic level of compliance to Microsoft's DDE specification. Microsoft has also defined a faster data format called XL_Table. Microsoft has only implemented this data format in Excel.
>
> FastDDE is a proprietary format used by Wonderware that is faster than XL_Table, but does not incorporate error checking.

From a data throughput standpoint, CF_Text is slow and XL_Table is fast. In a general way, you can think of CF_Text as an ASCII transfer with each data item being a single message, while XL_Table blocks the data into a table, so many items may be a single message.

RSLinx can communicate using any of these data formats. When a DDE channel is established, a form of handshaking takes place that automatically establishes the data format to be used (the fastest format both the client and server can understand).

AdvanceDDE

In addition to supporting CF_Text and XL_Table, Rockwell Software has also implemented its own DDE data format. This format is called **AdvanceDDE** (also known as PackedDDE).

AdvanceDDE is specifically designed for industrial communications giving the communications increased data throughput and well defined error checking to ensure data communications integrity.

AdvanceDDE's speed performance is accomplished in numerous ways. Among these, it allows for the binary transfer of data (thus there is no performance penalty for converting numeric binary data to text and back again). It also packetizes data to allow multiple data items per message (instead of one as in CF_Text).

All Rockwell Software Products can understand AdvanceDDE allowing for fast and accurate DDE communications.

Lab/Discussion: DDE Terminology

You should be familiar with the following terms (if not, please read the immediately preceding pages titled *DDE and how it works*).

- Clients
- Servers
- Client/Servers

Three Basic Parts to any DDE request:

- Application
- Topic
- Item

Examples	RSLinx	Excel
Application	RSLinx	Excel
Topic	Testsol (a PLC station)	[book1.xls]sheet1 (an Excel spreadsheet)
Item	T4:0.acc (a valid PLC data table address)	R2C3 (a cell address)

Types of Links:

Hot Link A hot link is a DDE channel that remains open, though it is only updated when the server sees a data change.

Cold Link A cold link is only opened upon command, such as via an Excel macro, or some other program implementing a DDE communication.

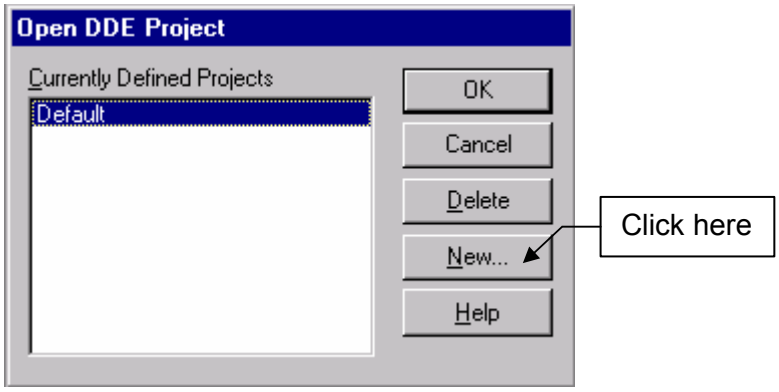
Types of links to RSLinx:

Bit/Word Read	Hot link or Cold link with a macro
Block Word Read	Hot link or Cold link with a macro
Bit/Word Write	Cold link with macro only
Block Word Write	Cold link with macro only

Training Section 6: Project, Topic and Alias Configuration

After configuring your communications hardware and verifying its connection, you can now configure a ***DDE/OPC Project and Topic*** to enable RSLinx to provide data to DDE requests. The Topic contains the information necessary to communicate with the intended PLC.

1. Click the *File* menu/*Open Project*, then click the New button.
   Type in a project name, such as *TEST* and click *OK*.



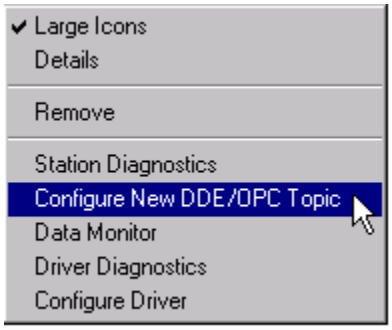
> ***Note**: If you have DDE connections open to RSLinx and select the Open Project menu item, the following message displays:*
>
> *"Cannot change projects while DDE connections are active to RSLinx. Please close all DDE connections to continue."*

Project names are limited to 49 characters. Spaces are converted to underscores (_). If the *Open Project* dialog box does not appear, go right to step two.

- When you create a project, you create a container for a number of topics; exactly how many topics is up to you and the amount of disk space you have available.
- It is usually best to add all the topics for a given scenario to a specific project. You may select to not create projects and use the default only, but there are advantages to grouping your topics into projects. Less of your computer's resources are used because instead of loading all of your topics, you might only load the ones you use for a specific acquisition or report.

Though you can create many projects, only one project can be used at a time.

2. A topic is a pointer to a PLC, much like how your telephone number “points” to your house. It also determines how fast or when data will be collected.
   Topics may be configured in two ways :



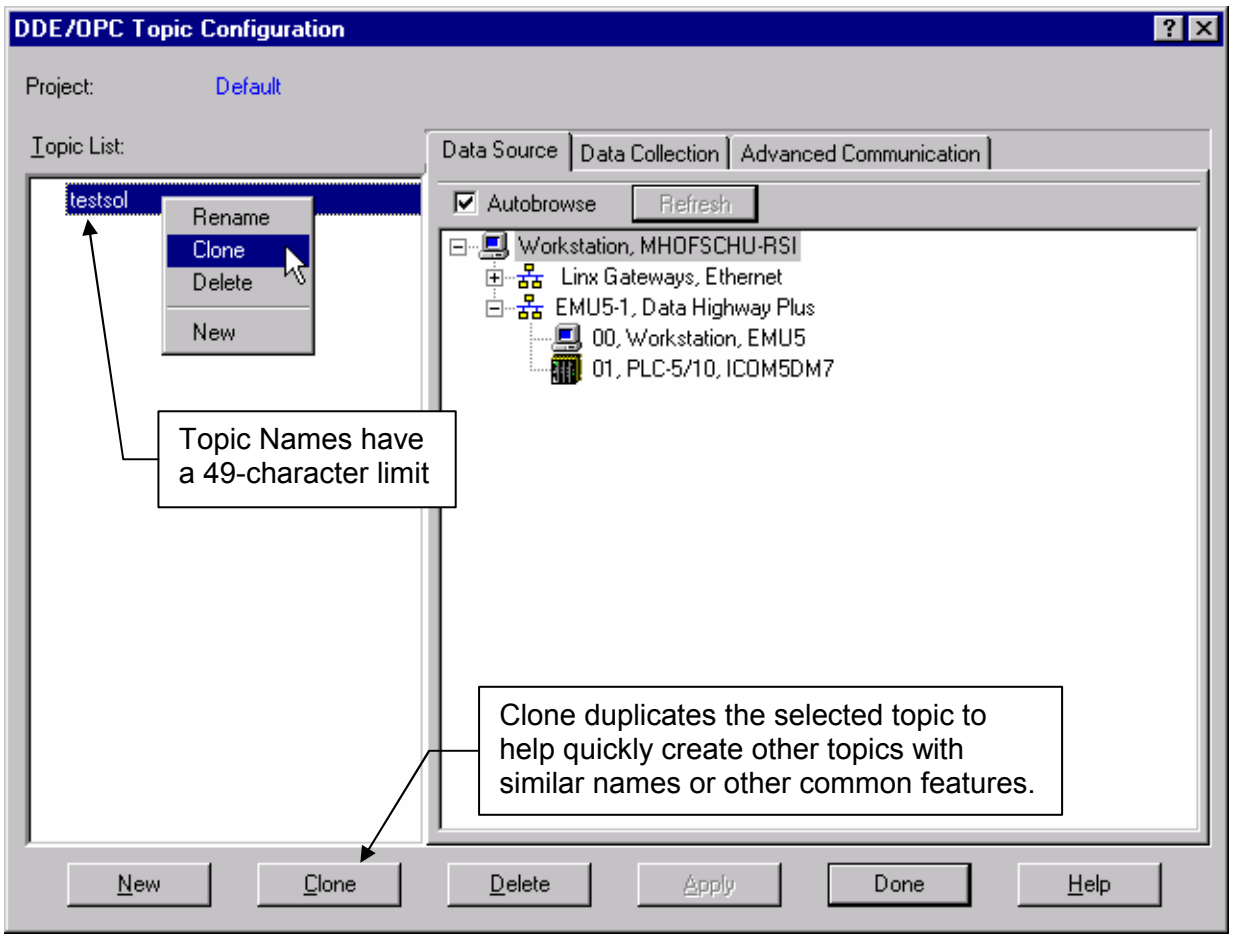
   - The preferred method is from an RSWho. Right-click a station and select “Configure New DDE/OPC Topic.”
   - This method will create a NEW topic – to only see or edit an existing topic, use the menu item called DDE/OPC - Topic Configuration…
   - Another way is to either use the Topic Configuration icon. 

   A DDE/OPC Topic Configuration dialog box displays. Click the help button in the dialog box for additional information on DDE topic configuration.

***Note**: Topic Configuration is not available with RSLinx Lite.*

The following Topic Configuration will appear:

DDE/OPC Topic Configuration – Data Source Tab



3. Replace the default topic name with the name **testsol**.
    - If the DDE/OPC Topic Configuration box is open by creating a topic from the RSWho, the following will be returned to this Topic Configuration box automatically:
        - The Station Address
        - PLC Family
        - Local/Remote addressing of the PLC
        - Communications Driver used to get to that station.

4. Click on the Data Collection Tab and set the Data Collection mode to Poll Rate and the rate at 100 mSec (one-tenth of a second).

--See illustrations on the next two pages.--

- *Polled Messages data collection mode instructs RSLinx to actively go out on the network and poll data. If you select Polled, you must also enter a polling rate in the Poll Rate (mSec) field. The minimum value for poll rate is 1ms, although actual speed depends on the number of active stations and data highway traffic.*
- *Unsolicited data collection mode* waits for a message instruction from the processor to send data to RSLinx.
- *Send all unsolicited updates* sends unsolicited data to the client regardless of whether or not the data has changed.

Use good judgement when selecting poll rates. Only collect fast changing data at the higher rates. Create slower topics to the same PLC for slower changing data. This technique can reduce network traffic if handled correctly.

Unsolicited data collection may be used alone or in conjunction with polling. It requires the PLC logic to be written to send a message to the RSLinx station. Unsolicited Message is an excellent way to reduce network traffic for infrequently changing data.

***Note**: It should be specified that this is an example for training purposes. In real applications, the poll rate should be set at a reasonable and/or realistic setting that is application dependent.*

The Limit Maximum Packets Option
Limit Maximum Packets on the Data Collection Tab is used in high load situations. It limits the maximum packets launched at any single point in time on a per topic basis. This helps the user not to overload a PLC station.

RSLinx allows values from 5-100 outstanding packets at one time.
Certain numbers, such as a PLC5 on a DH+ network, work best at less than 30 packets outstanding if only RSLinx is accessing the processor.

To determine the best value, change the pollrate to 1ms and do an advise on the item. If you receive an Out of Buffer Space message in the DDE Event Log, reduce the number of outstanding packets.

Limit Maximum Packets can be used to make updates more symmetrical for all optimized packets. The default value is 20, but for some PLCs and access methods, it may be necessary to change the default value for better performance and/or to reduce communication errors.

Collisions are possible when two products talk to the same PLC; therefore, you can only reduce, not eliminate, errors.

Update HotLink after a poke
Check this to update the DDE hotlink after a poke. This is a requirement of WonderWare.

Optimize poke packets
This option is only supported for PLC-5 and SLC processors, and in blockable DDE/OPC. DDE is only supported from FastDDE (WonderWare) and PackedDDE (AdvanceDDE) clients and from OPC clients.

- The entire packet must be received at once or it can not be optimized.

Keep DeviceNet connection open
Enabling this checkbox may increase DeviceNet Traffic. Do not check this box if the poll rate accuracy is greater than 5 seconds.

Enable this option if your poll rate exceeds 5 seconds and your desired poll rate accuracy is 5-10 seconds. The target device closes the connection if no communication occurs in 5 seconds.

- RSLinx will keep the explicit message connection open between the DDE server and the node configured in the topic.

Fail Unsolicited messages if data will be overwritten
Enable this box for the DDE/OPC client to receive all unsolicited messages, even if RSLinx must fail messages to ensure that no data is lost to the DDE/OPC client.

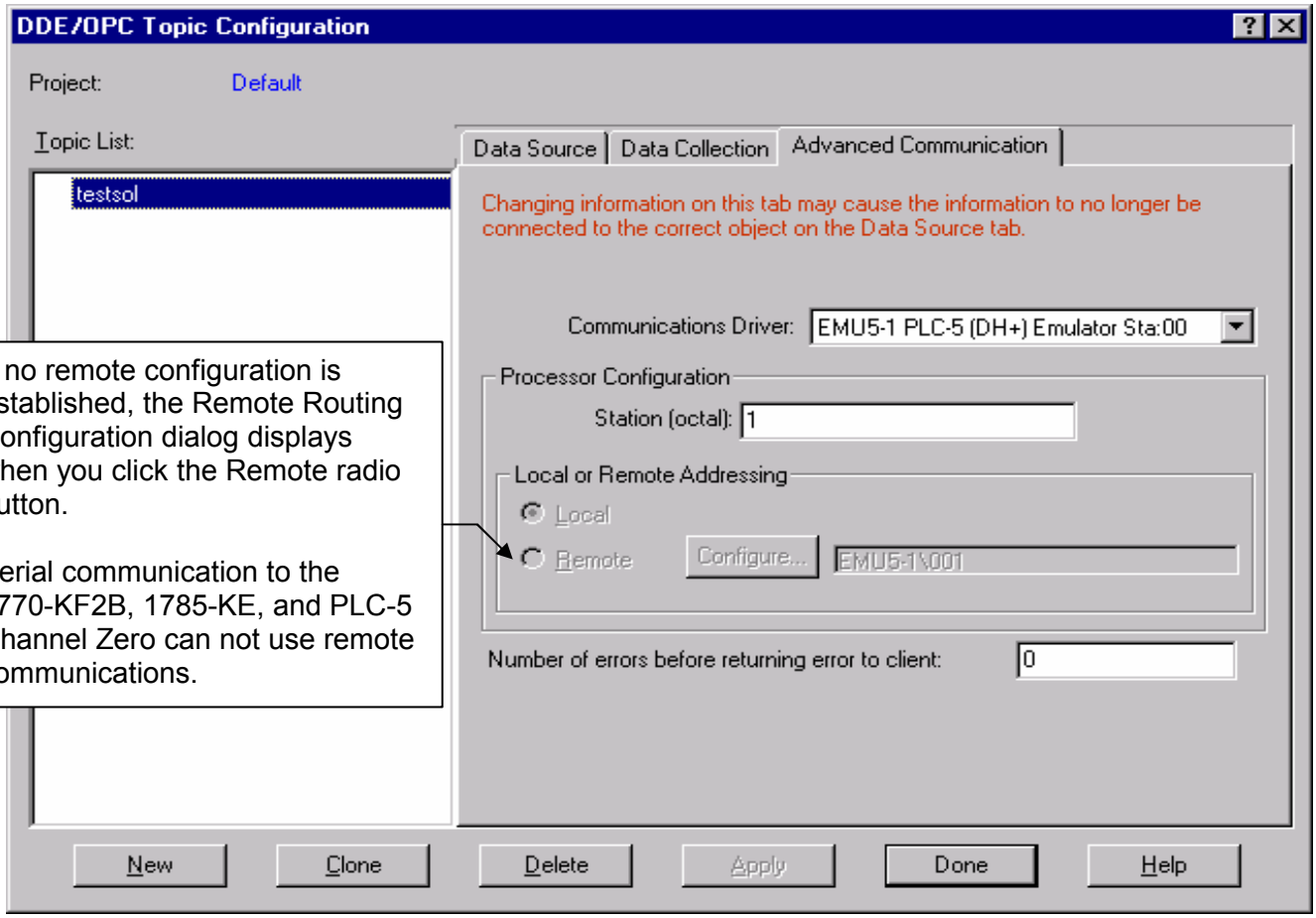
- This option is useful if the PLC sends too many unsolicited messages for the DDE/OPC client to handle, or the OPC client's group rate is slower that the PLC is sending unsolicited messages.

- If enabled, you must have ladder logic written in the PLC program to handle RSLinx failing the message for the PLC to resend the message.

Click the Advanced Communications Tab, look but do not change anything.

> ***Important:*** *Changing the information on this tab may cause the information to be disconnected from the correct object on the Data Source tab.*
>
> *This tab is usually only used to set up a new topic without starting from the RSWho. The RSWho new topic creation method automatically applies the driver and station information.*

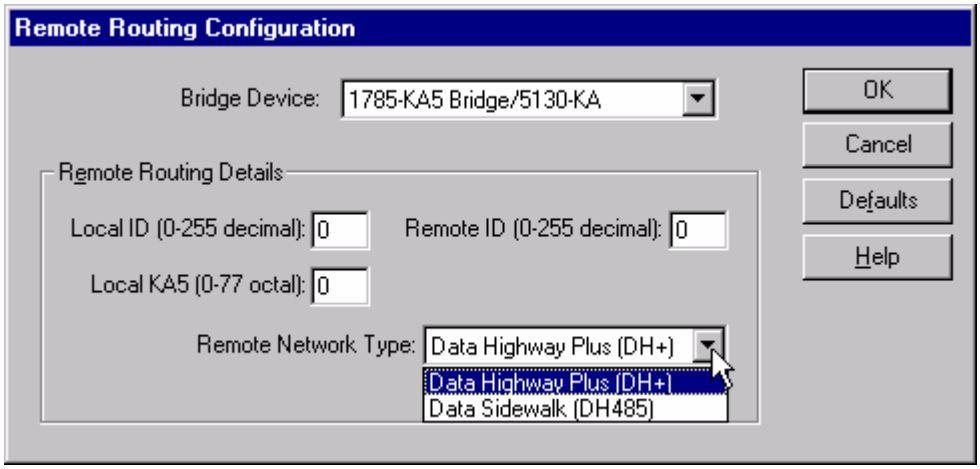
DDE/OPC Topic Configuration – Advanced Communication Tab



If no remote configuration is established, the Remote Routing Configuration dialog displays when you click the Remote radio button.

Serial communication to the 1770-KF2B, 1785-KE, and PLC-5 Channel Zero can not use remote communications.

Local or Remote Addressing
These options vary depending on the current communication hardware driver and the bridge device selected.



RSLinx supports six different methods (listed in the Bridge Device drop-down list box) for accessing remote processor networks:

- 1785-KA Local/Remote
- 1785-KA5 Bridge/5130-KA
- Ethernet EI RM Module
- Ethernet EI 5130-KA Module
- Remote ControlNet
- 1756-DHRIO

See RSLinx Help for specific instructions.

5. The topic configuration is now complete. Click the Done button

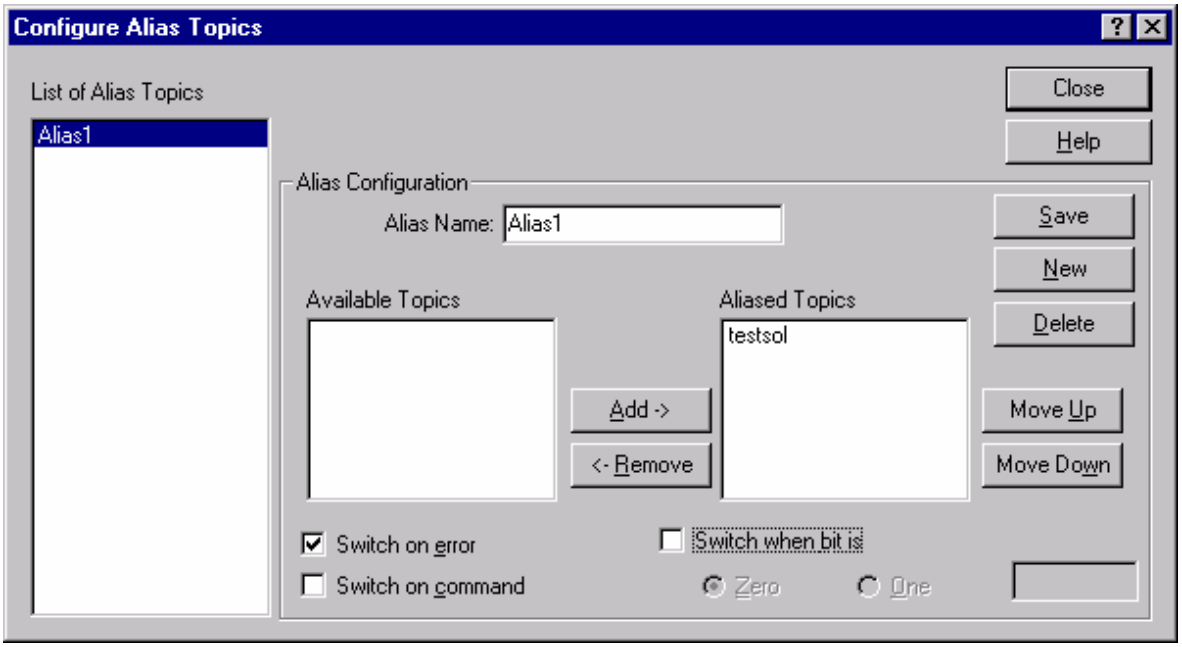
Alias Configuration

An Alias Topic is a “virtual” or “redundant” topic that you can use to reference other topics. This allows you to define a “virtual topic” which can be directed to one of several physical PLC devices, with automatic switchover.

You can also configure the criteria used for switching from primary to backup devices, or you can force a switchover using a DDE Execute command. Alias topics are not mandatory for communication to occur.

For example, if you have a PLC accessed by an Ethernet driver, a KT driver, and a DF1 driver, you could create three topics. You can then create an alias topic which prioritizes:

1) Ethernet driver's topic
2) KT driver's topic
3) DF1 driver's topic



The following information is displayed in the Alias Topic Configuration dialog box:

- **List of Alias Topics** shows a list of configured alias topics. If a topic is active and is selected, all configuration items for that topic will be grayed out.

- **Alias Name** is the reference to the alias topic.

- **Available Topics** shows a list of device topics available for the alias topic. Once an aliased topic is used, only matching topics will be displayed.

- **Aliased Topics** shows a list of topics in the order to be switched to on configured switch parameters.

- **Switch on Error** tells the alias topic that any error that occurs on the topic causes an immediate switch to the next topic in the aliased topic list. If it is presently active on the last topic, it switches back to the beginning.

- **Switch on Command** tells the alias topic to allow a DDE Execute to switch the topic to the next or another topic.

- **Switch when bit is** (either “zero” or “one” are the options) tells the alias topic that if the watched bit goes to zero, switch the alias topic to the next aliased topic (see Switch on error).

EXAMPLE: Set the active DDE topic configured in an alias topic

This example displays how to programmatically set the active DDE topic that is configured in an alias topic using Microsoft Excel VBA.

```
RSIChan = DDEInitiate("RSLinx", "System")
Application.DDEExecute RSIChan, "[Set_Alias(My_Alias,KT_Topic)]"
```

where:

- RSLinx is the DDE application name.
- System is a predefined topic of RSLinx.
- My_Alias is the user configured alias name in RSLinx.
- KT_Topic is the DDE topic that RSLinx switches to

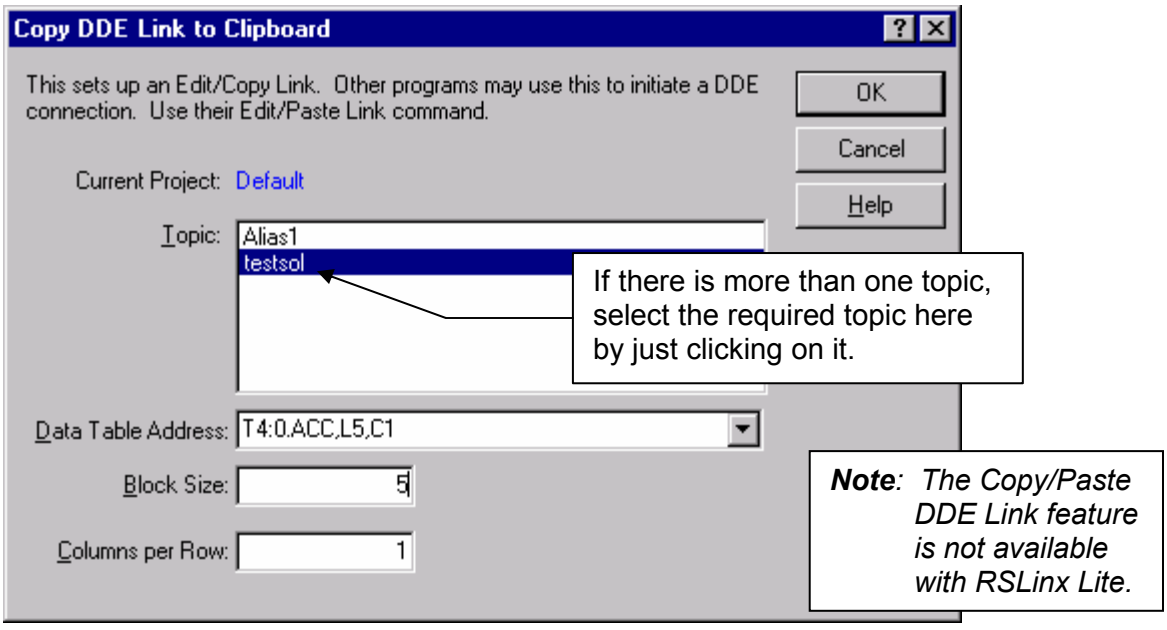
Training Section 7: Establishing a DDE Link to Microsoft Excel

Once a DDE topic is configured, you are ready to establish a ***DDE Hot Link*** to Microsoft Excel. There are two ways to establish a DDE Hot Link to Excel, the hard way and the easy way. The hard way is to type it by hand with all the proper syntax marks in place, but we will try it the easy way using the Windows clipboard. The clipboard will copy *only* hot links.

This exercise will establish a Hot Link to Excel to read a block of five timer accumulators from a PLC-5 and create an active graph of the data in Excel. We will use T4:0 through T4:4 - it is essential that these timers exist in the data table section of the PLC-5. Excel 97 is used for this exercise, other versions of Excel would be similar with slightly different keystrokes or mouse clicks.

Lab/Discussion:

1. If RSLinx is not already running, start it now.
2. Select *Edit/Copy DDE Link to Clipboard* (or use the icon in the toolbar that looks similar to the typical windows Paste icon.)
   - A Copy Link to Clipboard box will appear.



3. The Data Table Address you wish to copy goes in the first block, in this case, we'll use the default address: *T4:0.ACC*.

4. In the Block Size box enter a **5**. This is the length of the block copy that will be made, five timer accumulators starting at address T4:0.ACC.

5. In the Columns per Row box leave it at the default value of **1**. This will format the block of data to be placed in one column in the destination application which is Microsoft Excel in this case

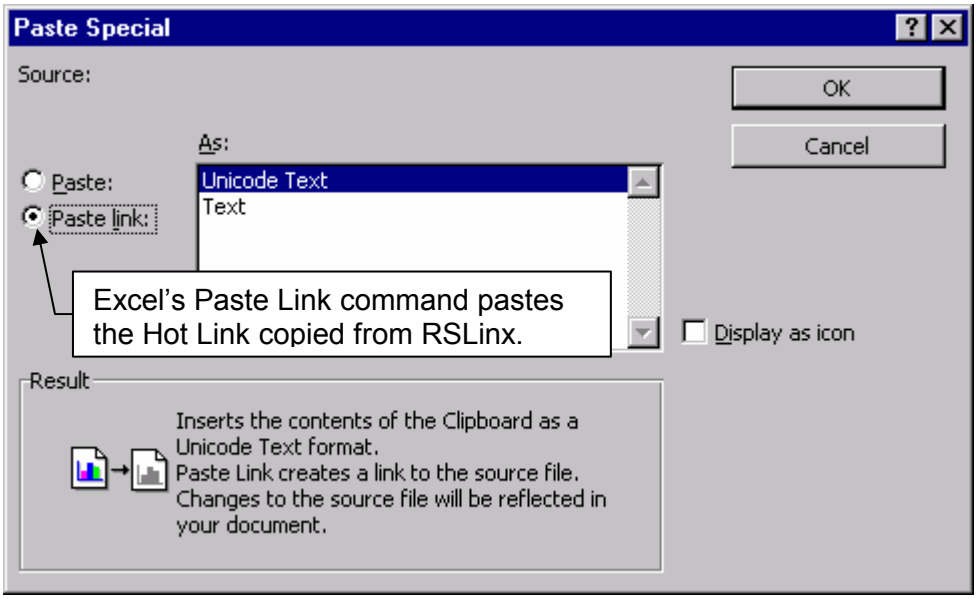
6. Pick the topic you have configured, ***TESTSOL***, which is displayed in the *Select a Topic* box.

7. Click *OK*. All of the information required to establish a Hot Link to Excel is now residing in the Windows clipboard.

8. Start-up Microsoft Excel by clicking the taskbar's Start button/ Programs/Microsoft Excel, but DO NOT CLOSE RSLinx.

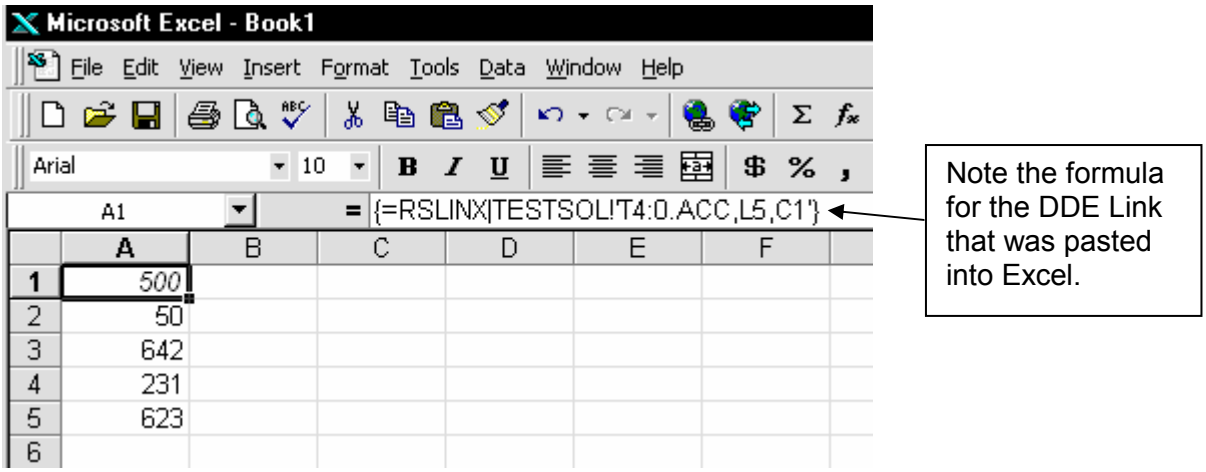
9. Place the cell pointer in the cell that you would like the Hot Link to be put in. For example, place the cell pointer in cell A1 by RIGHT-clicking on that cell.

10. Select *Paste Special*, source of *Paste Link* and click on *OK.* The Hot Link will be pasted into the cell and data from the PLC will appear.
    - The five timer accumulator values will begin to update in five successive cells. This is now live data being read directly from the PLC-5.

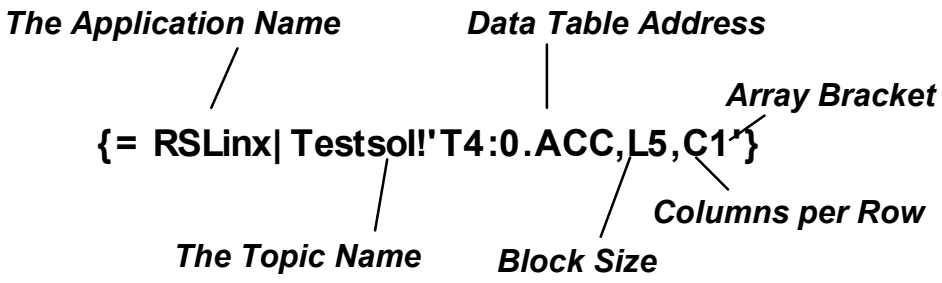


11. Note that the formula bar in Excel contains the formula that is needed to establish the DDE Hot Link. This link was pasted in from the clipboard.

    It should look similar to the following:



12. The parts of the formula are as follows:



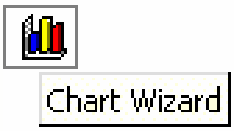
Note the braces { } around the formula. These indicate that this formula is an *array* formula. All hot links pasted in via the clipboard from RSLinx are array formulas and are READ ONLY (in other words - you can not poke, also known as “download,” from a hot link).

Under normal circumstances, you cannot modify part of an array formula (an array is defined by Excel - please see your Excel manual for details).

- If you attempt to do this you can get caught in a loop of errors. To bail out, just press the escape key on your keyboard.

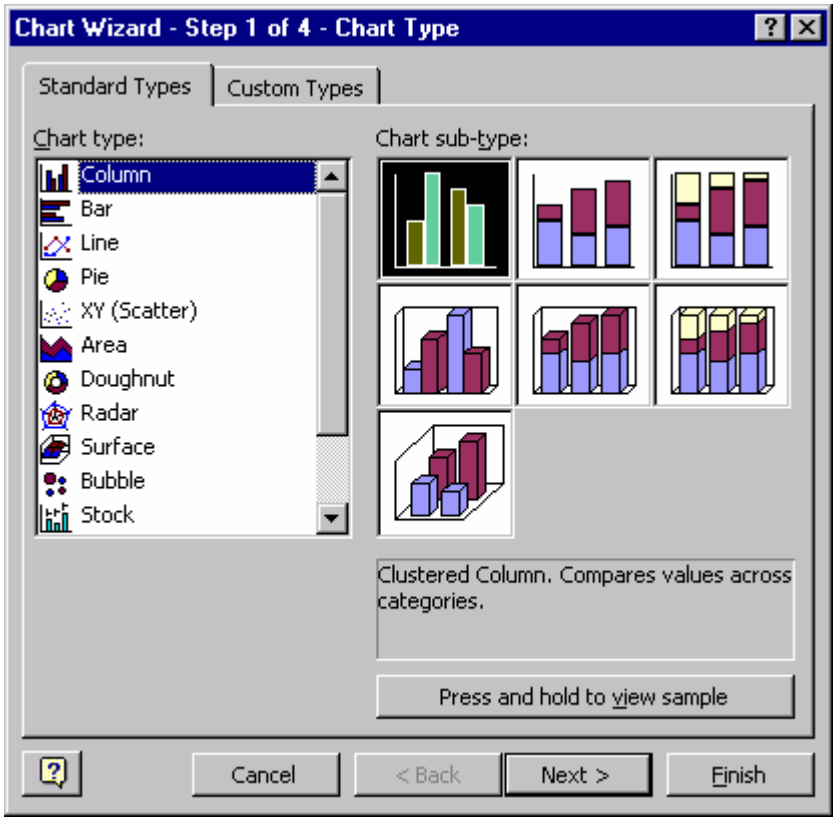
Charting Data In Excel

13. We are now going to create a live chart from the PLC data. Make sure that the five cells that contain the PLC data are highlighted (Left-click on cell A1, hold, and drag down to cell A5).
14. Click on the Chart Wizard icon on the Excel toolbar. The Chart Wizard icon looks like the following:



Excel 97
Chart Wizard
Icon & Tooltip

15. The following Chart Wizard Step 1 of 4 will immediately appear as the Chart Wizard icon is clicked.
    - You will see a box to select the type of chart you prefer.



***Note**: At this point, you may simply accept the Wizard defaults for these options.*

16. After selecting the chart type, just click Finish or, if you prefer, click Next repeatedly until the Wizard is complete just to be aware of the charting options available.

17. After clicking **Finish,** a live chart of the data will appear on your spreadsheet.

18. Click on the File menu/Save As… and call the new workbook Test.xls and save it in the default folder.

19. The exercise is complete.

Training Section 8: Writing a Simple Microsoft Excel Macro

Goal: In this section, we will go through the steps to write a macro that will write a piece of data to a register in a PLC. The macro will be controlled by a typical windows button.

We will write a value to a timer preset. This lab assumes that you have completed section six and that both RSLinx and Excel are active in your Windows environment. It assumes that you have a valid working topic named "testsol" (connected to either a PLC-5 or an emulated PLC-5).

> ***Note**: It is very important that the following steps are taken exactly as described. If one character is typed incorrectly, your macro may not work.*

Lab / Discussion:

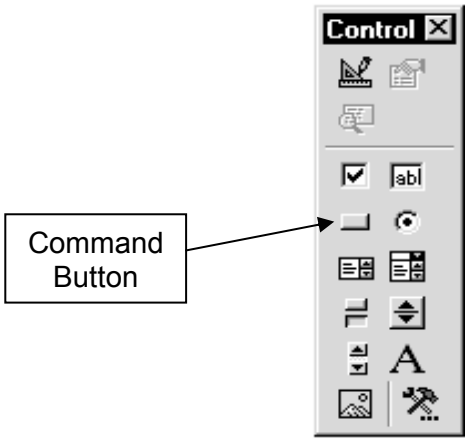
1. If Excel is still running, close it and do not save changes - or if you wish to save your changes, save them under a new name. Do not save them under the default name of Book1.

2. Note: **DO NOT CLOSE RSLinx** itself, only close Excel!

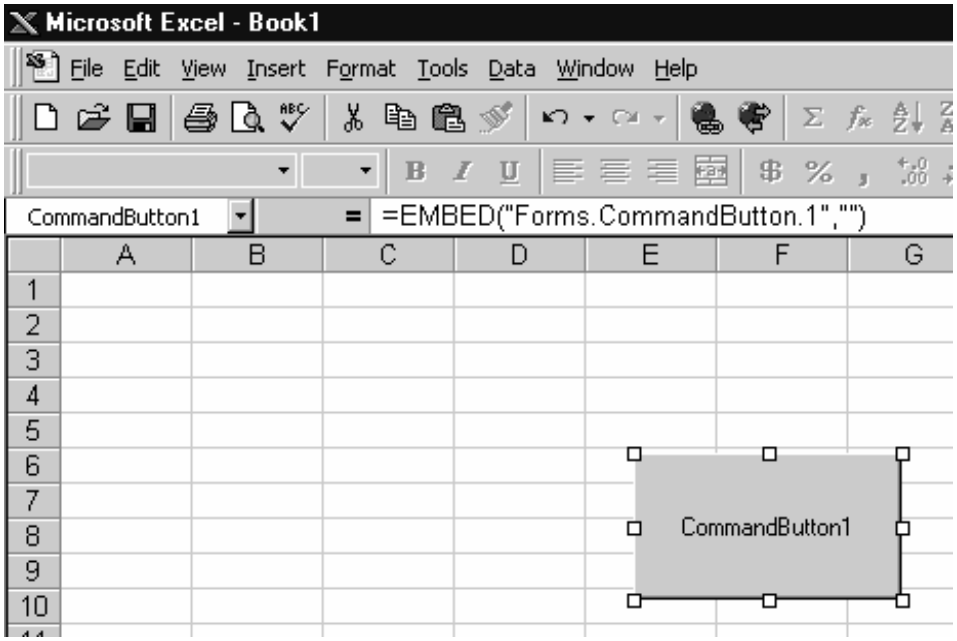
3. Start Excel again. Excel should open to a blank workbook named *Book1*, with a blank spreadsheet named *Sheet1* (you will see a tab in the lower left-hand corner of your screen named Sheet1).

4. Rename the sheet you are looking at to "DDE_Sheet" by double clicking on the "Sheet1" tab and entering the new name of DDE_Sheet.

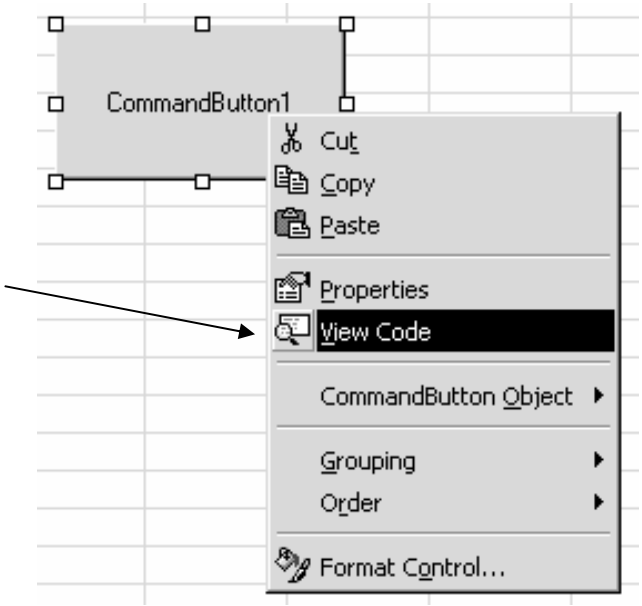
5. Next, we will create a button. From the View pulldown menu, choose Toolbars/Control Toolbox. A small toolbar should appear on your spreadsheet. It should look like the following example:



6. Click on the Button tool and the mouse pointer will turn into a cross. Now you must identify where you would like to place the button on your sheet
   - For the purpose of this exercise, place the button somewhere to the right of Column D.
   - With your "cross" cursor, click-hold and drag your mouse to outline the size, shape and location of your button. Your Excel spreadsheet and button should look similar to the following:

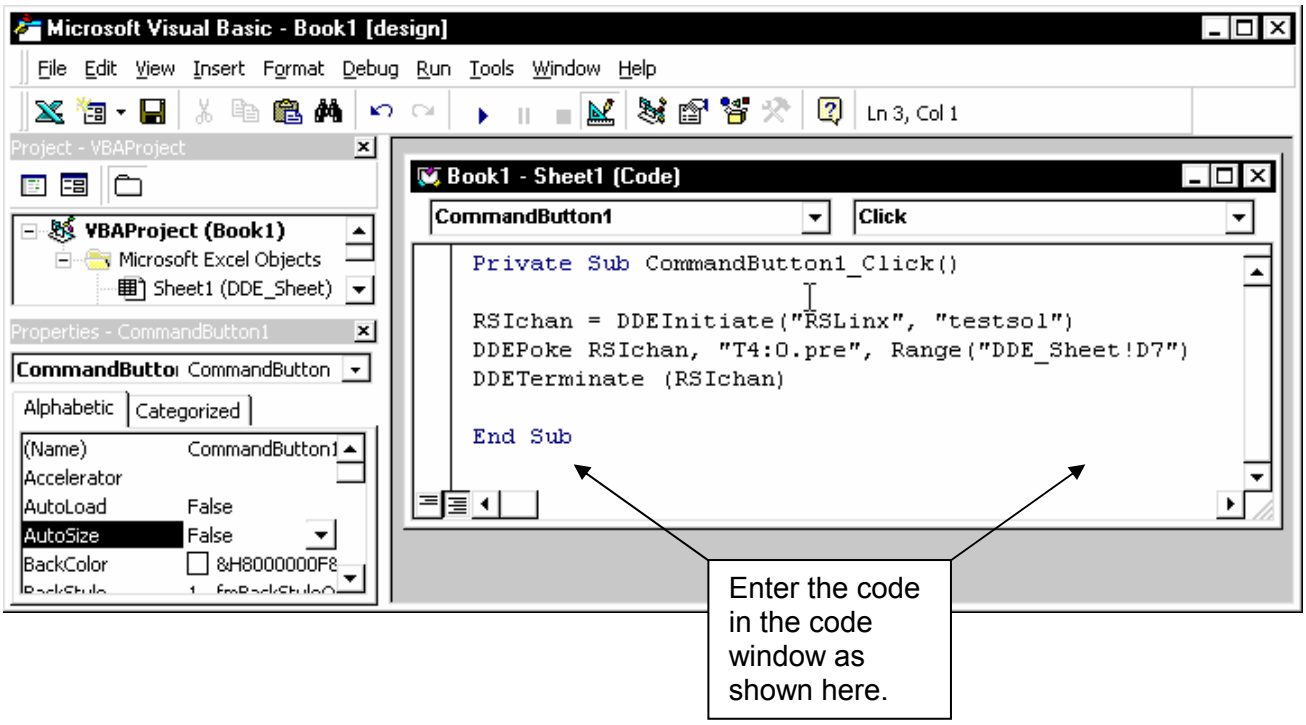


7. Right click on your new button and choose *View Code* as shown below.



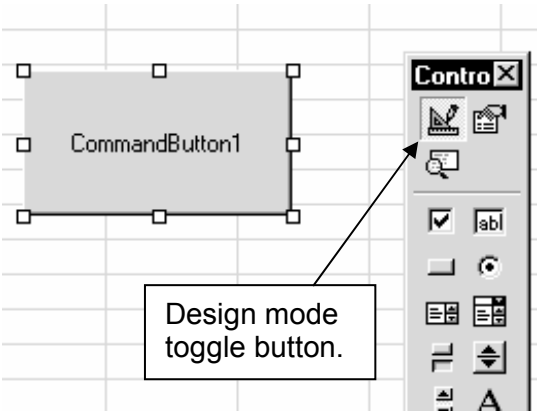
8. The View Code option opens the Excel 97 Visual Basic environment.
   - From here you will enter the code (subroutine) which will execute when you push the button. Enter the following code into the code window as shown below:

     RSIchan = DDEInitiate("RSLinx", "testsol")
     DDEPoke RSIchan, "T4:0.pre", Range("DDE_Sheet!D7")
     DDETerminate (RSIchan)



9. Return to the normal Excel spreadsheet window by single clicking on the Excel icon or Excel tab on the task bar at the bottom of your screen.

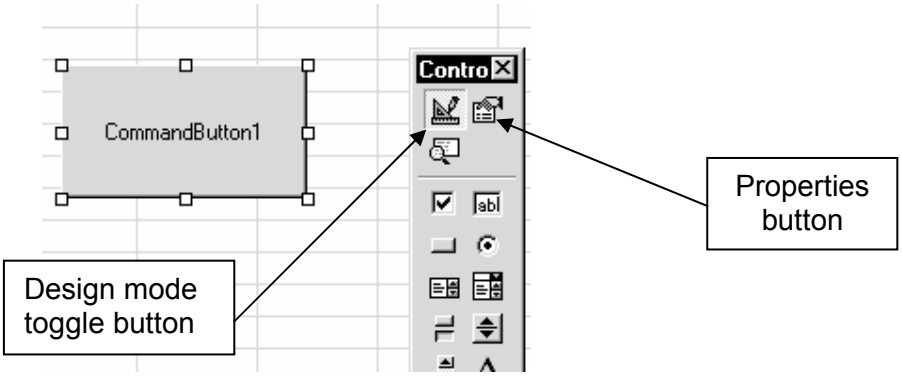
10. Your command button is still in design mode, and needs to be put into normal run mode. To do this, click on the design mode tool on the Control Toolbox as shown below.



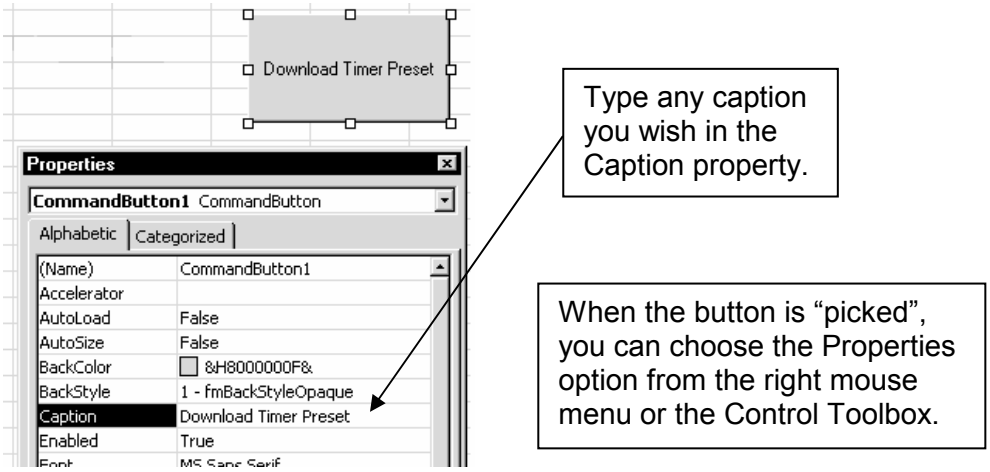
11. Your command button should now be ready to function. However, before you press it, you should put a value into cell D7.
    - This is the location that the macro code you wrote expects to find the value that it will write to T4:0.PRE.
    - Type 300 into cell D7 (be sure you press [**Enter**] to have Excel accept the number into the cell). Then click your button.

12. To verify that the new value was indeed written to the PLC, you could add a hot link into the Excel spreadsheet that reads the value from T4:0.PRE.

13. You can change the text on the button by changing the *caption* property of the button. To do so, you must put the button back into design mode.
    - Press the Design Mode toggle button on the Control Toolbox.



14. To access the properties for the button, you can right-click on the button and select the Properties option. When the properties window appears, find the Caption option in the list of properties and type in an appropriate title for your button.



Additional Macro Examples:

Using procedures similar to those on the previous pages, the following Visual Basic Code can be added to additional command buttons to do a read from PLC, a block read, and a block write.

Read a word (reads T4:0.acc and places in cell C7 on **DDE_Sheet**):

```
RSIchan = DDEInitiate("RSLinx", "testsol")
data = DDERequest(RSIchan, "T4:0.acc")
Range("DDE_Sheet!C7").Formula = data
DDETerminate (RSIchan)
```

Write a block of five words (from cells A7-A11 on **DDE_Sheet**):

```
RSIchan = DDEInitiate("RSLinx", "testsol")
DDEPoke RSIchan, "N7:30,L5", Range("DDE_Sheet!A7:A11")
DDETerminate (RSIchan)
```

Read a block of five words (and displays in cells B7-B11 on **DDE_Sheet**):

```
RSIchan = DDEInitiate("RSLinx", "testsol")
data = DDERequest(RSIchan, "N7:30,L5,C1")
Range("DDE_Sheet!B7:B11").Formula = data
DDETerminate (RSIchan)
```

ROCKWELL SOFTWARE

Training Section 9: Local and Remote OPC

Introduction

This section outlines step by step the configuration of Windows to use OPC, local and remote, with Rockwell Software products. Both Windows 95/98 and NT are covered along with RSLinx, RSView32 and RSTools.

The steps outlined here are the results of actual experimentation with Windows and RSI products. There may be other ways to configure Windows for OPC over a network, but these are the steps that Rockwell Software has found to be successful.

OPC on one PC

***Note**: OPC is not available with RSLinx Lite.*

Configuring Windows

OPC between two applications on the same PC uses windows COM (Component Object Model). No special configuration of COM or windows is required if only one PC is involved. However, if you are using Windows 95, you will have to install DCOM (Distributed Component Object Model) for Windows 95 before you can use COM.

- DCOM does not ship as part of Windows 95. A self-extracting executable can be found on http://www.Microsoft.com/com/dcom95 . Download and run it. DCOM installs as part of Windows 98 and Windows NT 4.0 or higher so this step is not necessary.

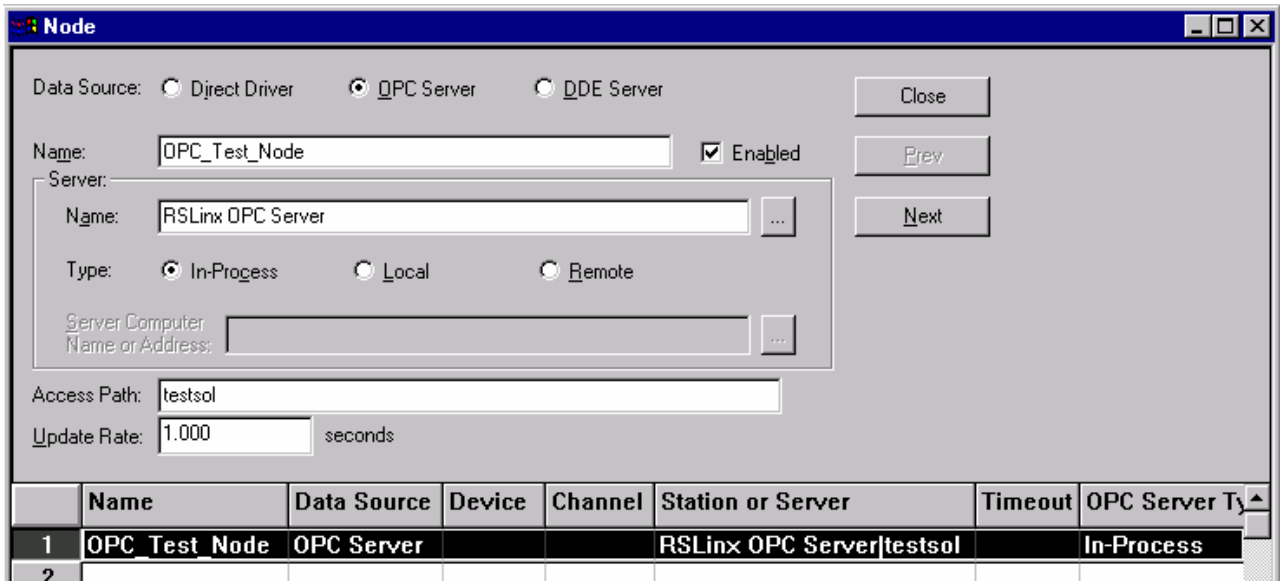
Configuring Rockwell Software Products

No special configuration should be required for Rockwell Software Inc. (RSI) servers built with the RSServer OPC Toolkit such as RSLinx or other RSServer products. You must have RSLinx 2.0 or higher to use OPC. See the help files of each specific RSServer to see if it supports OPC or call Rockwell Software.

Configuration of OPC varies from client to client. RSView32 and RSTools will be covered here. See the help files of other RSI client product for instructions on configuring OPC or call Rockwell Software.

RSView32 6.0 or higher can be configured for OPC through the node configuration. Under the systems folder in the project dialog, choose “Node”. Radio buttons on the top of the dialog will allow you to select Direct Driver, DDE or OPC as a node type. Choose OPC.

Next, move to the “Name” text box. Enter a name for the station. It can be anything you want. Then move to the “Server Name” text box, click the browse button . Select the OPC server you wish to connect to, “RSLinx OPC Server”
for example. The “In-Process” radio button will be automatically selected.

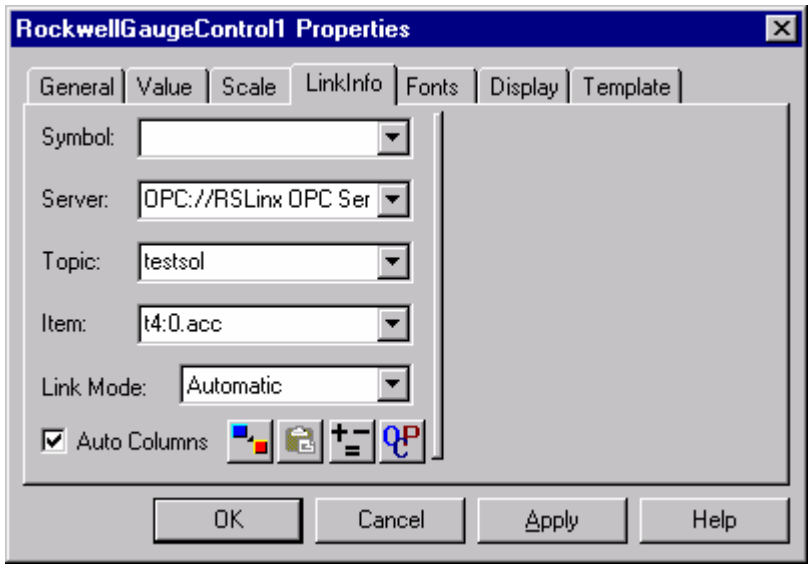


Finally under “Access Path”, type in the name of the DDE topic you wish to connect to in the server. In other words, “Access Path” is the DDE topic. Note, that although you are entering a DDE topic name, RSView32 will communicate to the server via OPC, not DDE! That’s it!

RSTools can connect to an OPC server by using a special syntax for the application name under the “Link” Tab. Right click on the RSTools Active X control and select properties.

Go to the “Link” tab. Under application type in “OPC://Server Name”, for example “OPC://RSLinx OPC Server”. Enter the topic name as usual.

Note: *You must have a version of tools that specifically supports OPC. See the help file or call Rockwell Software to see if you have the right version.*



OPC over a network

***Notes**: (1) To use OPC over a network with RSLinx, you must have RSLinx Gateway 2.0 or higher. RSLinx OEM or regular RSLinx will not do OPC over a network.*

(2) Remote OPC works best when RSLinx is run as a service.

- *When running RSLinx as a service, **do not** open it as an application!*

Configuring Windows for DCOM

OPC between two PCs over a network uses Windows DCOM. The setup is slightly different depending on the version of Windows you are running. Depending on the server and client implementations of OPC, you may only have to make the following configuration changes on the server. *In the case of most RSI applications, these changes should be made on the server and the client PC.*

Win 95/98

DCOM does not ship as part of Windows 95 so it must be installed. DCOM for Windows 95 must be installed on both PCs. A self-extracting executable can be found on http://www.Microsoft.com/com/dcom95. Download and run it.
DCOM installs as part of Windows 98 so this step is not necessary. Next, *you must have a domain controller on the network for WIN95/98 DCOM to work!*

Setting up a Primary Domain Controller.

A Primary Domain Controller is a computer in a Windows NT Server Domain that authenticates domain logons and maintains the directory database for a domain. The primary domain controller tracks changes made to accounts of all computers on a domain. It is the only computer to receive the changes directly. A domain has only one primary domain controller.

First, you must determine if you have a domain controller on your network. Under Windows 95 or 98 go to control panel and open Network Properties. Under the configuration tab select Client for Microsoft Networks and click the Properties button. If Login to Windows NT Domain is checked then you are probably on a network with a domain controller. If so you can skip to the DCOM configuration section.. Check with your network administrator if you are unsure.

If you do NOT have a domain controller on your network, you will need to setup a PC running Windows NT Server (not workstation) as a domain controller.
Don't do this on your office or plant network without checking with your network administrator first.

A primary domain controller is configured during the Windows NT Server (not workstation) install. If you are upgrading an existing NT install, you must choose to do a new install of NT, not the upgrade when prompted. Just install over the existing NT directory. The setup program will prompt you to choose between a Primary Domain Controller, a Backup Domain Controller or a Stand Alone Server. Chose Primary Domain Controller and enter a name for your Domain when prompted. Then continue with the install as you normally would.

Once NT Server is installed as a Domain Controller then go to the start menu, choose administrative tools, and User Manager for Domains. This application will allow the Domain Administrator (YOU!) to add users to your newly created domain. Add accounts for all your users to your domain. Next go to the start menu, administrative tools, and choose Server Manager. Add accounts for all PCs you want in the domain. This adds the actual machine (PC) name to the domain, not the user.

Now you are ready to go configure DCOM on your PCs that will be running Rockwell Software. Since DCOM uses Windows security, there are several ways to enable a DCOM connection between two PCs

Allow some or all domain users privileges under DCOM Configuration.
In order to do this Access control must be set to "user level" first under the access control tab of network configuration. Go to Network Neighborhood and right click. Choose properties and go to the access control tab. Select "user level access control." You must fill in a domain name in the text box. Enter the name of the domain from which you wish to allow users access You will have to reboot at this point.

Next go to the start menu and choose run. Type in "DCOMCNFG" and click OK. Go to the Default Security tab and click "edit default". Click add and choose the domain from where you want to add users. Choose "Everyone" from the list. This will grant all domain users DCOM access to this Windows 95 or 98 system.

- As an alternative, you may choose to add only specific users that you want to grant access to this PC.

Also on this tab, check the "Enable remote connection" check box at the bottom of the dialog. Finally, go to the Default Properties tab and check the "Enable Distributed COM on this computer" check box.

The above method allows users to access any COM enabled application on the PC. The most secure method is to grant users rights only to specific applications. This is also done through the DCOMCNFG application.

Allow some or all domain users privileges to a specific application only under DCOM configuration.
In order to do this Access control must be set to “user level” first under the access control tab of network configuration. Go to Network Neighborhood and right click. Choose properties and go to the access control tab. Select “user level access control.” You must fill in a domain name in the text box. Enter the name of the domain from which you wish to allow users access. You will have to reboot at this point.

Next go to the start menu and choose run. Type in “DCOMCNFG” and click OK. Go to the Applications tab and choose the application you want to grant access to. Click the properties button and go to the Security tab. Choose “use custom access permissions” and click edit.

Click add and choose the domain from where you want to add users. Choose “Everyone” from the list. This will grant all domain users DCOM access to this Windows 95 or 98 system.

- As an alternative, you may choose to add only specific users that you want to grant access to this PC.

Next go to the default security tab and check the “Enable remote connection” check box. Finally, go to the Default Properties tab and check the “Enable Distributed COM on this computer” check box.

WIN NT
DCOM should install as part of Windows NT 4.0 or higher.

Since DCOM uses Windows NT security, there are several ways to enable a connection between two PCs.

Use the same login on both PCs. The simplest way is to log into both PCs using the same login. A domain login can be used like \\domain\username or if using workgroups a local account can be used like “username”.

Since using the same login on the server and client PCs may not be practical, there are several other ways to enable DCOM between two PCs. *All of these methods require that a Primary Domain Controller exist on the network.*

Allow some or all domain users privileges under DCOM Configuration.
Go to the start menu and choose run. Type in “DCOMCNFG” and click OK. Go to the Default Security tab and click the first “edit default” button (the one under the Default Access Permissions section). Click add and choose the domain from where you want to add users. Choose “Everyone” from the list. This will grant all domain users DCOM access to this Windows NT system.

- As an alternative, you may choose to add only specific users that you want to grant access to this PC.

Finally, go to the Default Properties tab and check the “Enable Distributed COM on this computer” check box.

The above method allows users to access any COM enabled application on the PC. A more secure method is to grant users rights only to specific applications. This is also done through the DCOMCNFG application.

Allow some or all domain users privileges to a specific application only under DCOM configuration.
Go to the start menu and choose run. Type in “DCOMCNFG” and click OK. Go to the Applications tab and choose the application you want to grant access to. Click the properties button and go to the Security tab. Choose “use custom access permissions and click edit. ”. Click add and choose the domain from where you want to add users. Choose “Everyone” from the list. This will grant all domain users DCOM access to this Windows NT system.

- As an alternative, you may choose to add only specific users that you want to grant access to this PC.

Finally, go to the Default Properties tab and check the “Enable Distributed COM on this computer” check box.

Configuring Rockwell Software for OPC over a network
No special configuration should be required for RSI servers built with the RSServer OPC Toolkit such as RSLinx or RSServers. You must have RSLinx 2.0 or higher to use OPC. See the help files of each specific RSServer to see if it supports OPC or call Rockwell Software. However, because Rockwell Software clients use *local* registry entries to find information about OPC servers, the server must be registered on both the client and server PCs.

The easiest way to do this is to install the server on all PCs involved, clients and servers. Activation for the server does not have to be installed, since the server will not be run on the client machines, just the registry entries are required.

Another way to register the server is by using a ".REG" file. Double clicking on a file with the .REG extension will insert the contents of that file onto the PCs registry. This file may be available from the Rockwell Software, call for more information. The OPC foundation is considering standardizing the way a client might browse a server machine over the network for a list of available OPC servers. Such a standard would eliminate the need to register the server on the client PCs. Rockwell Software will implement this feature once the OPC foundation has agreed on a standard method.

Configuration of OPC via DCOM varies from client to client. RSView32 and RSTools will be covered here. See the help files of other RSI client product for instructions on configuring OPC or all Rockwell Software.

RSView32 version 6.0 or higher can be configured for OPC through the Node Configuration. Under the Systems folder in the project dialog, choose "Node ". Radio buttons on the top of the dialog will allow you to select Direct Driver, DDE or OPC as a node type. Choose OPC. Next move to the Name text box and enter a name for the station. It can be anything you want. Then move to the server text box, click browse. Select the OPC server you wish to connect to, RSLinx OPC Server for example. Below this text box select the "Remote" radio button. In the Server Computer Name or Address text box enter the name of the server PC or click browse if you are on a domain and select the server PC. Finally, under Access Path type in the name of the DDE topic you wish to connect to in the server. The Access Path is the DDE topic. Note, that although you are entering a DDE topic name, RSView32 will communicate to the server via OPC, not DDE!

RSTools can connect to an OPC server by using a special syntax for the application name under the "Link" Tab. Right click on the RSTools Active X control and select properties. Go to the "Link" tab. Under application type in "OPC:\\Server PC Name/Server Name", for example "OPC:\\Server\RSLinx OPC Server". Enter the topic name as usual.

Note: *You must have a version of tools that specifically supports OPC. See the help file or call Rockwell Software to see if you have the right version.*

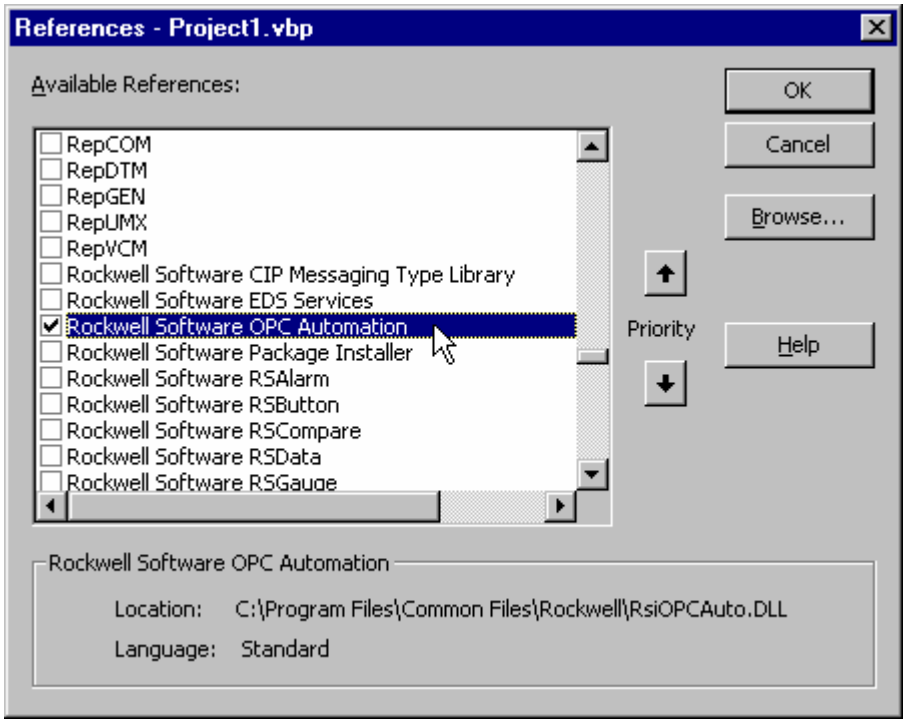
Training Section 10: Using OPC Automation in VB to Read a Single Element

- The Basics -

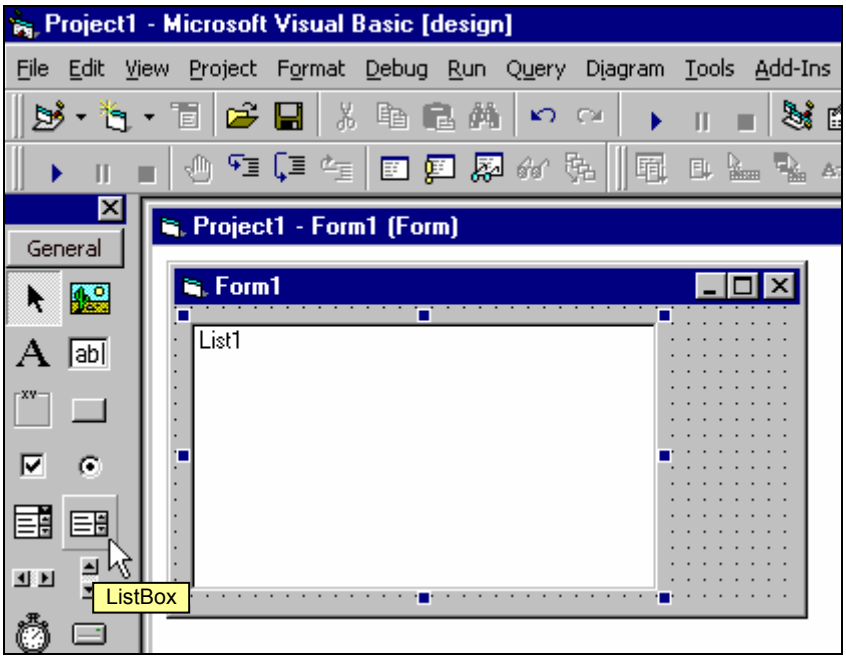
1. Open **Microsoft Visual Basic** and create a new **Standard EXE** project [Start] > Programs > Microsoft Visual Studio > Microsoft Visual Basic

2. From the main Visual Basic menu, click **Project** and **References**.

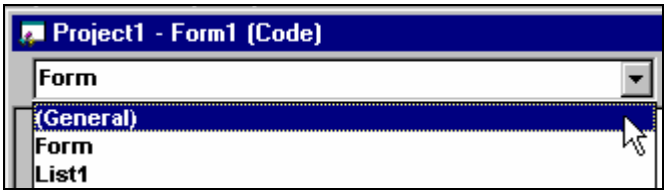
3. Select **Rockwell Software OPC Automation** from the list of Available References and click OK



4. Add a **ListBox** control to **Form1** as shown.

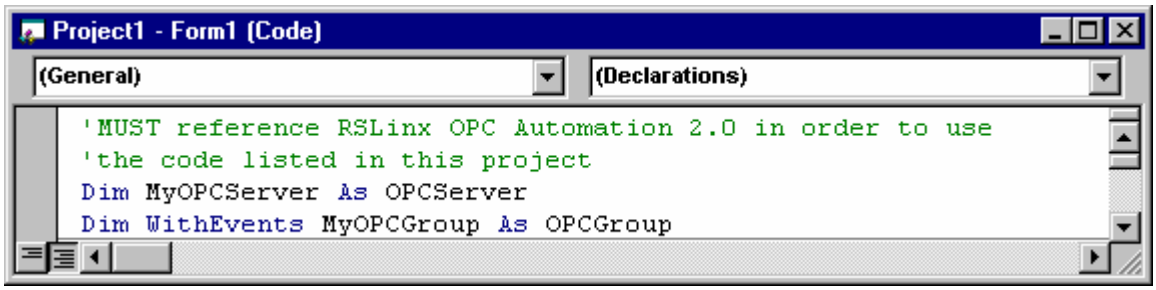


5. Double click in a gray area of **Form1** to display the code window for the project. Click the drop down list of Objects (Currently displaying the word **Form**) and select **General**.

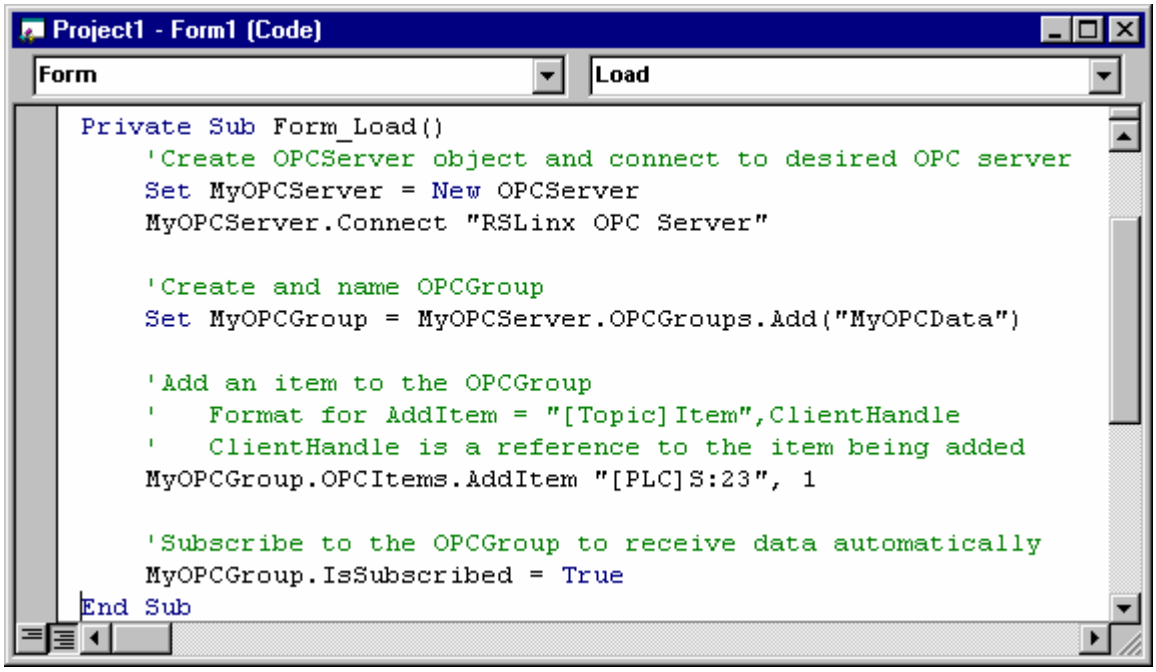


6. Enter the following lines of code to the **[General][Declarations]** section of the project.

   Note: Throughout this lab any line of code starting with an apostrophe (‘) is a comment used to document the code. These lines are not required for the project to run properly and do not need to be entered.

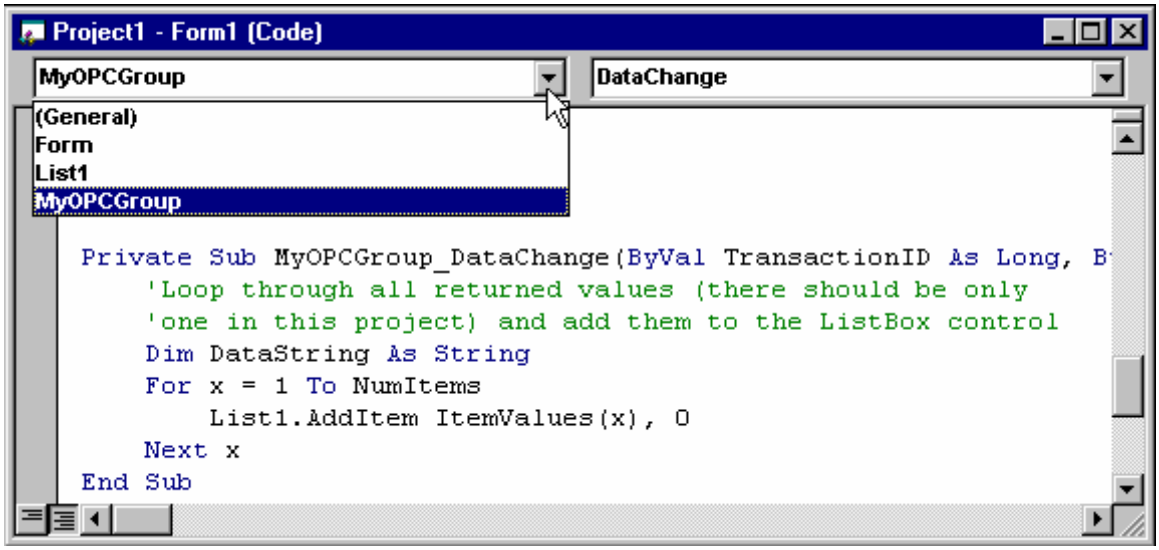


7. Add the following code to the **Form_Load** event.

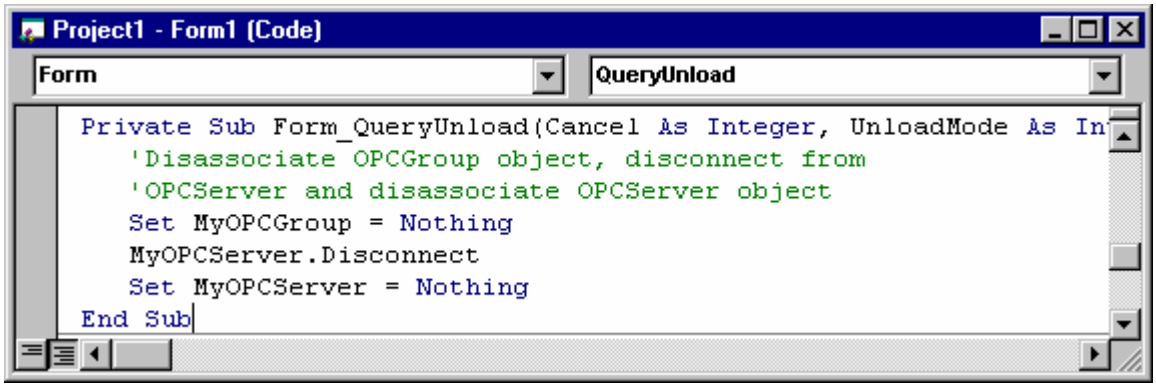


8. Click on the drop down list of objects and select MyOPCGroup. The **DataChange** event will be displayed by default. Add the following code to the **MyOPCGroup_DataChange** event.

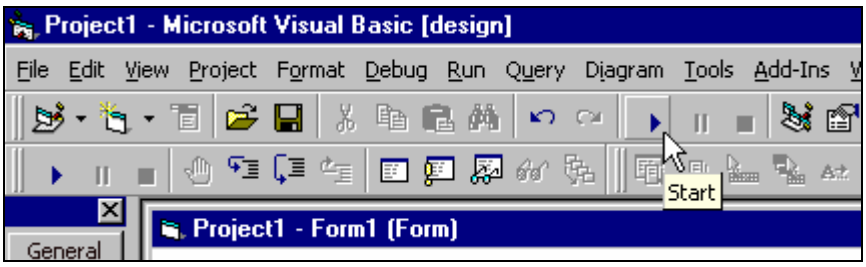
   Note: If MyOPCGroup is not listed in the drop down list of objects, review steps 2-7 and verify each one was performed accurately.

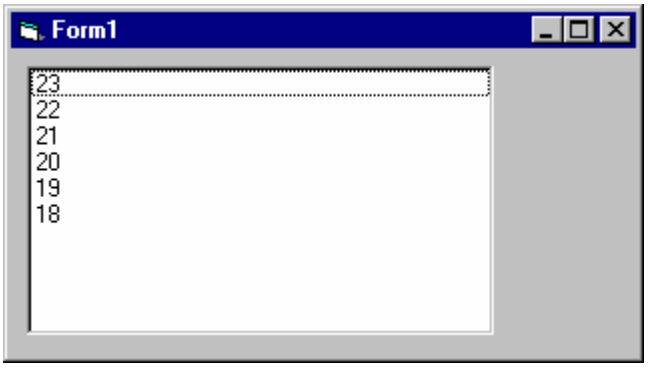


9. Add the following code to the **Form_QueryUnload** event.



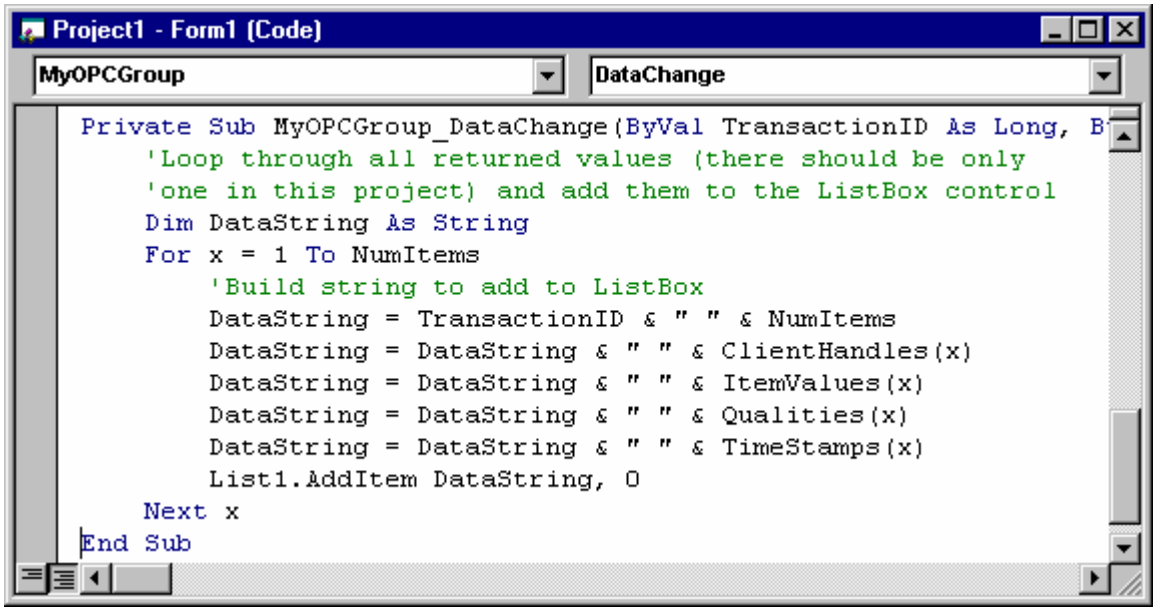
10. Start the project and observe the changing data.





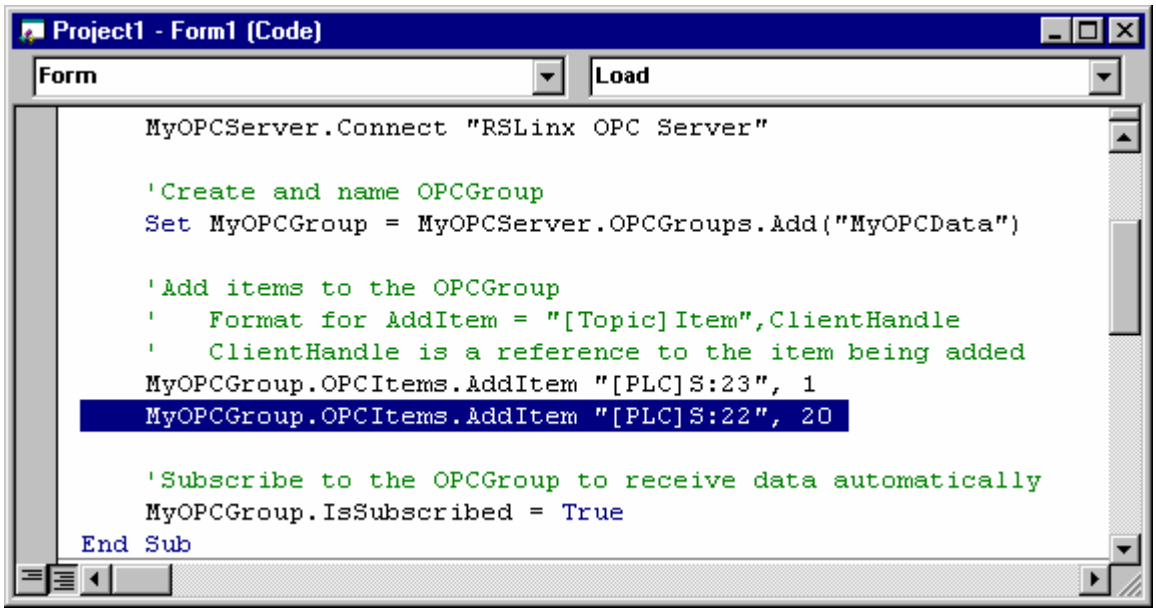
- Going Further -

11. Display all of the information returned by the OPC Server in the ListBox by modifying the **MyOPCGroup_DataChange** event code as follows.



Start the project and observe the values being returned for the TransactionID, ClientHandle, ItemValue, Quality and TimeStamp.

12. Add another data item to monitor by modifying the Form_Load event as follows. Run the project and watch closely to observe 2nd data point.



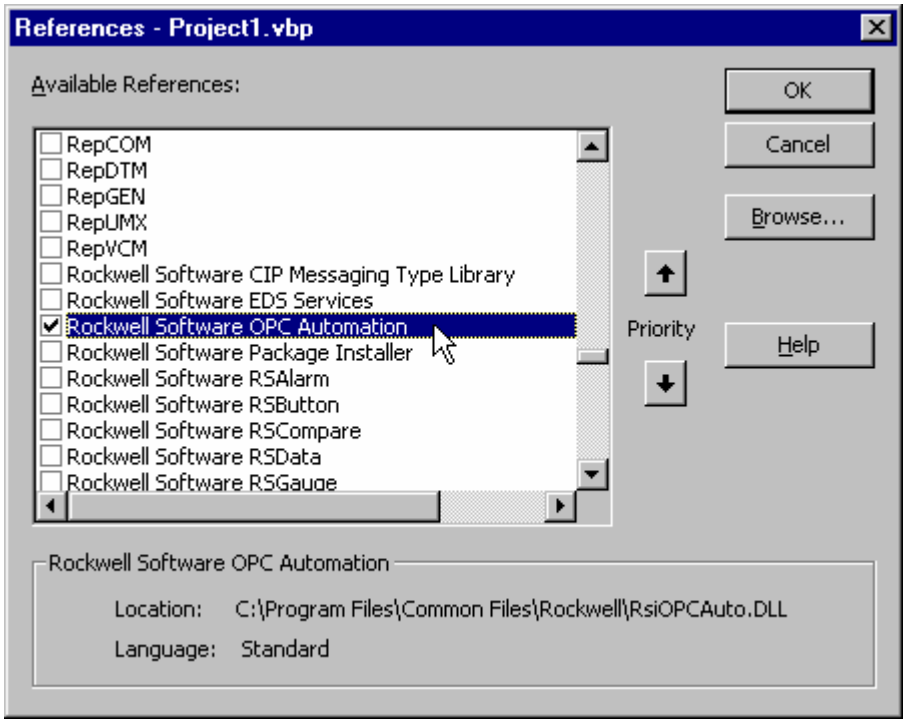
Training Section 11: Using OPC Automation in VB to Write a Single Element

- The Basics -

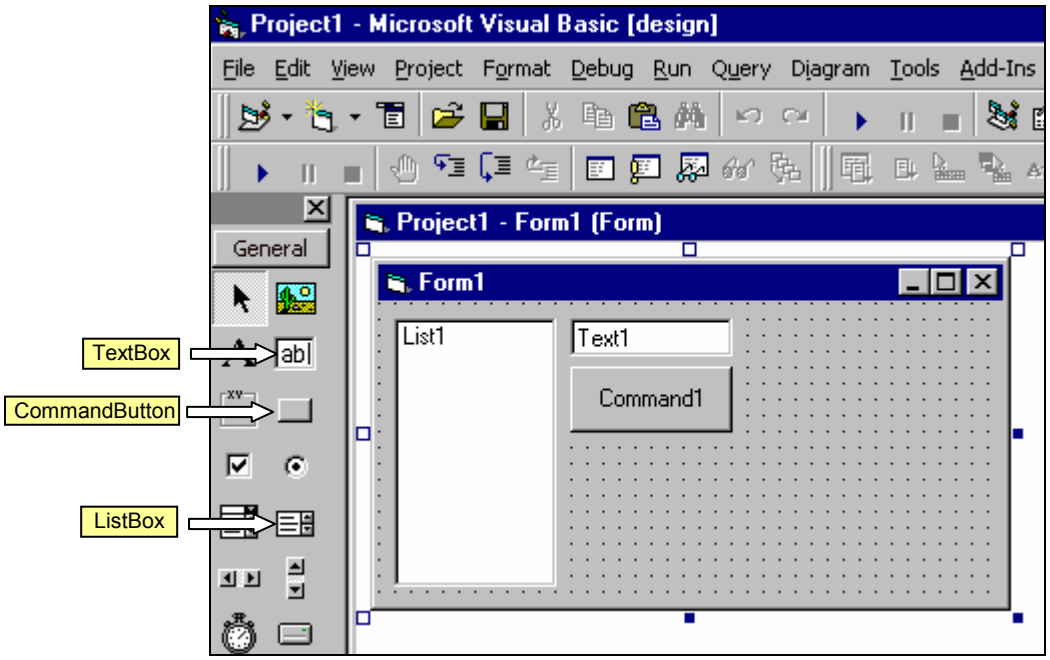
13. Open **Microsoft Visual Basic** and create a new **Standard EXE** project
    [Start] > Programs > Microsoft Visual Studio > Microsoft Visual Basic

14. From the main Visual Basic menu, click **Project** and **References**.

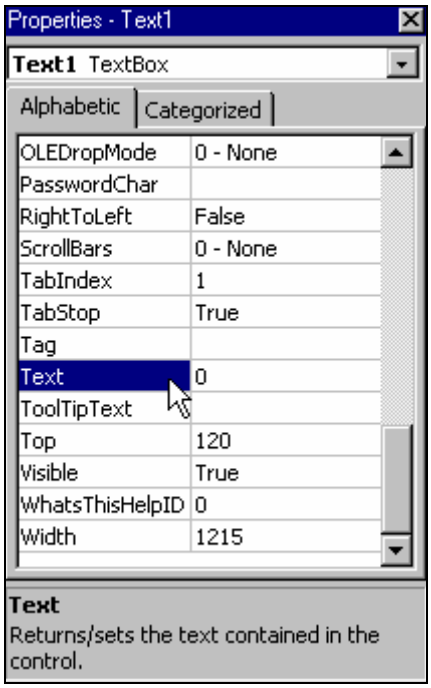
15. Select **Rockwell Software OPC Automation** from the list of Available References and click OK



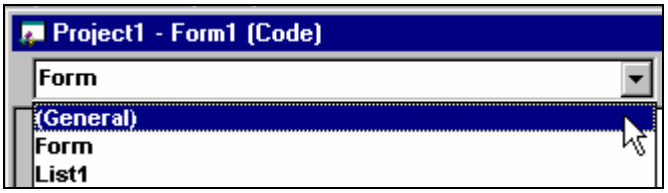
16. Add a **ListBox**, a **TextBox** and a **CommandButton** to **Form1** as shown.



17. Click on **Text1** to give the TextBox object focus. Using the **Properties** dialog box on the right side of the screen, change the Text property to zero (0).

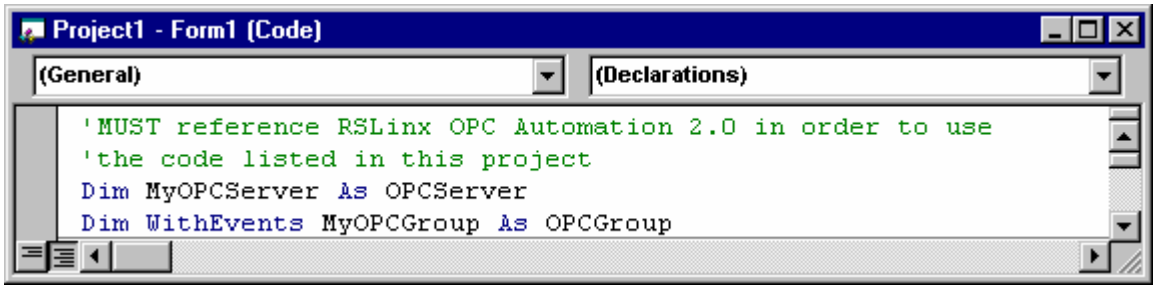


18. Double click in a gray area of **Form1** to display the code window for the project. Click the drop down list of Objects (Currently displaying the word **Form**) and select **General**.

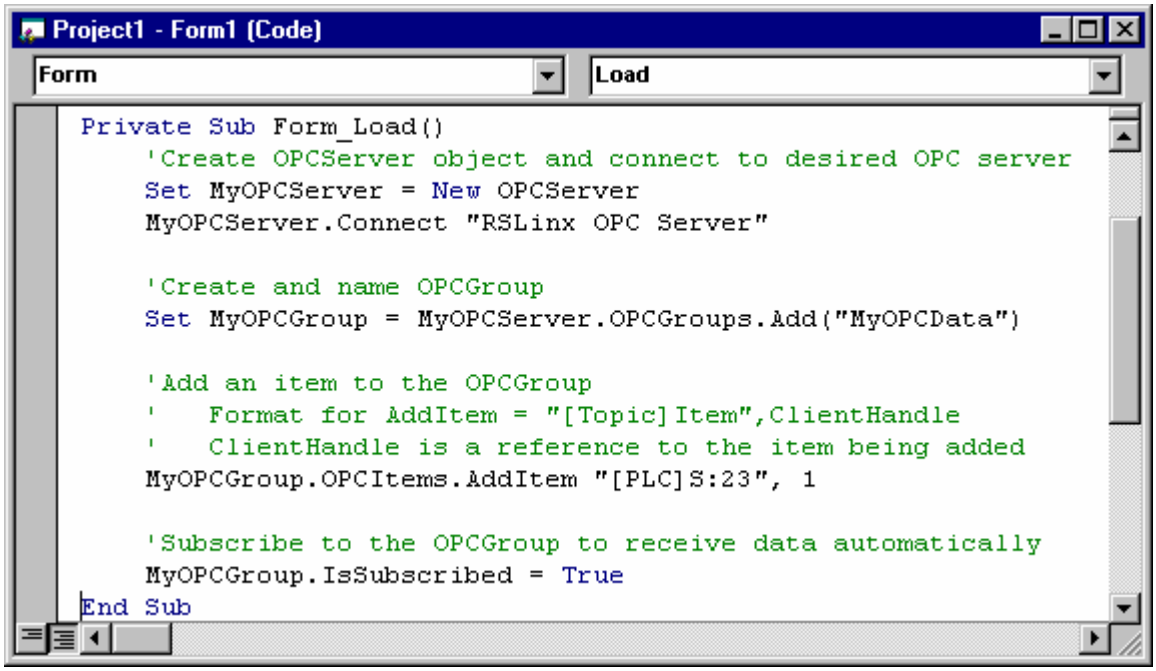


19. Enter the following lines of code to the **[General][Declarations]** section of the project.

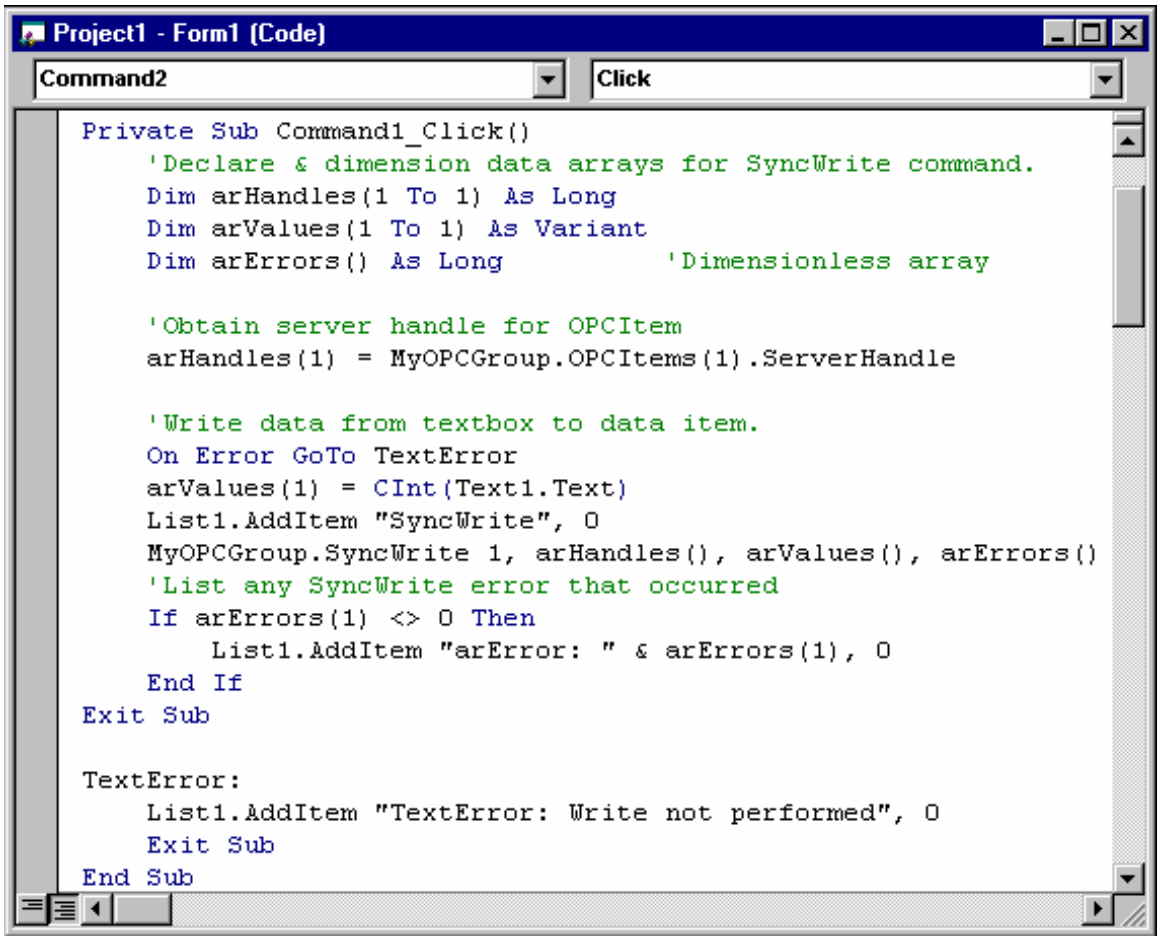
    Note: Throughout this lab any line of code starting with an apostrophe (') is a comment used to document the code. These lines are not required for the project to run properly and do not need to be entered.



20. Add the following code to the **Form_Load** event.

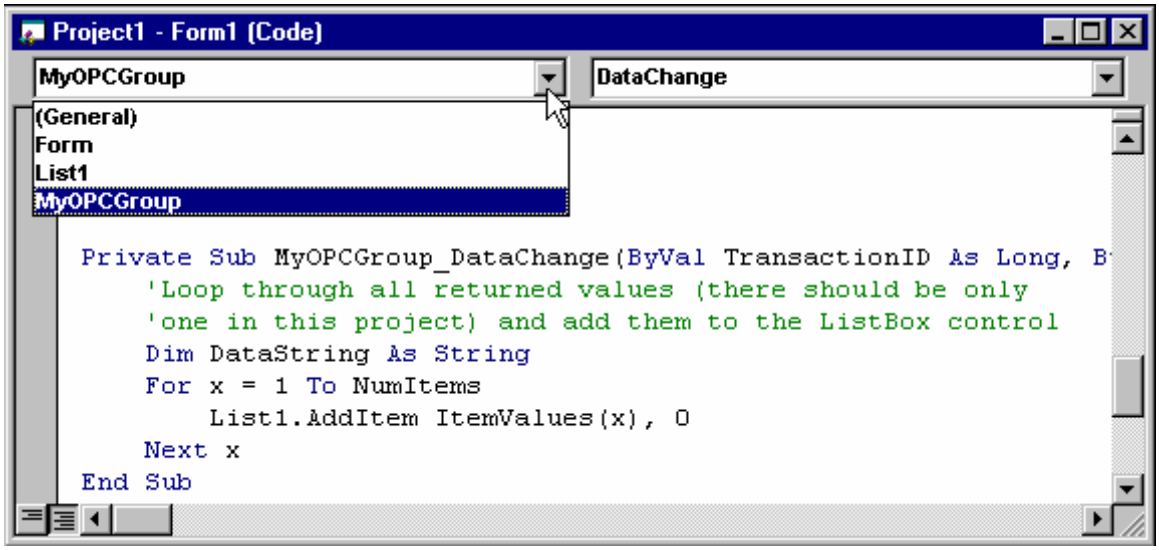


21. Add the following code to the **Command1_Click** event. Note the dimensions for the **arHandles** and **arValues** arrays are appropriate for only one data item. Also note that the **arErrors** array remains dimensionless.

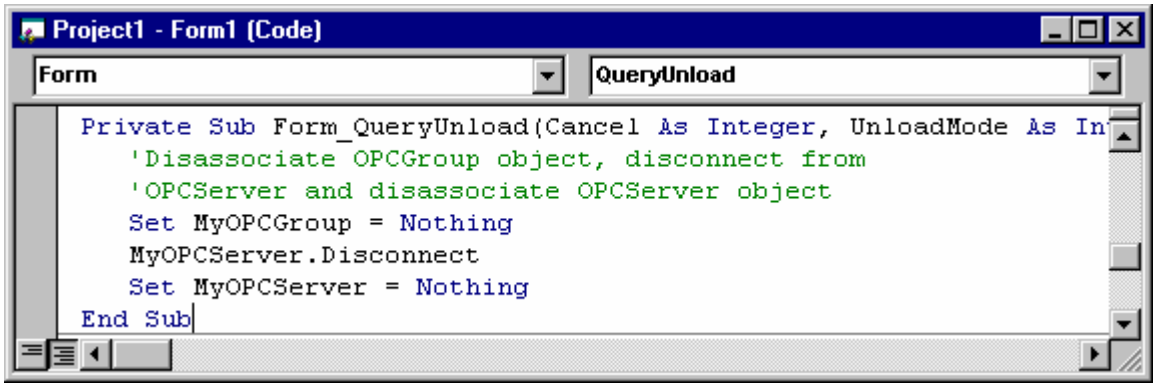


22. Click on the drop down list of objects and select MyOPCGroup. The **DataChange** event will be displayed by default. Add the following code to the **MyOPCGroup_DataChange** event.

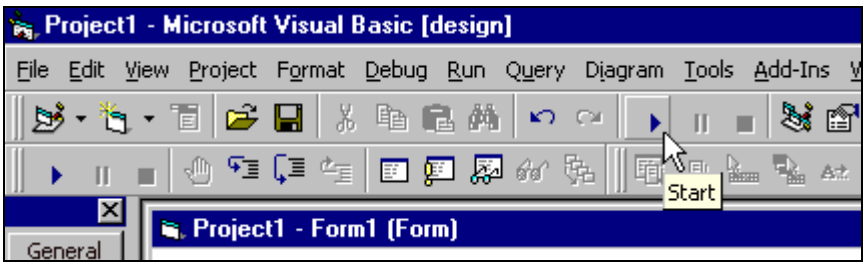
    Note: If MyOPCGroup is not listed in the drop down list of objects, review steps 2-7 and verify each one was performed accurately.



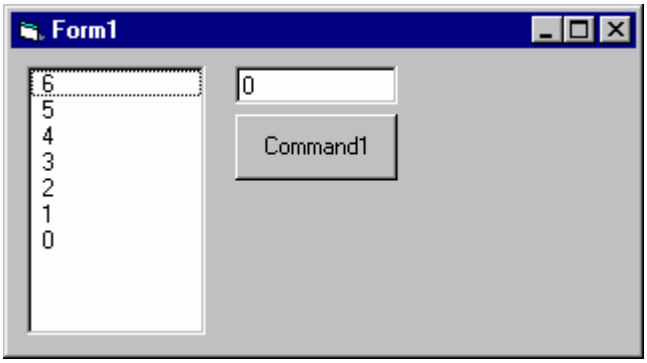
23. Add the following code to the **Form_QueryUnload** event.



24. Start the project and observe the changing data.



25. Enter a number from 0 to 60 in the TextBox and click the Command1 button. Observe how the data item is affected.

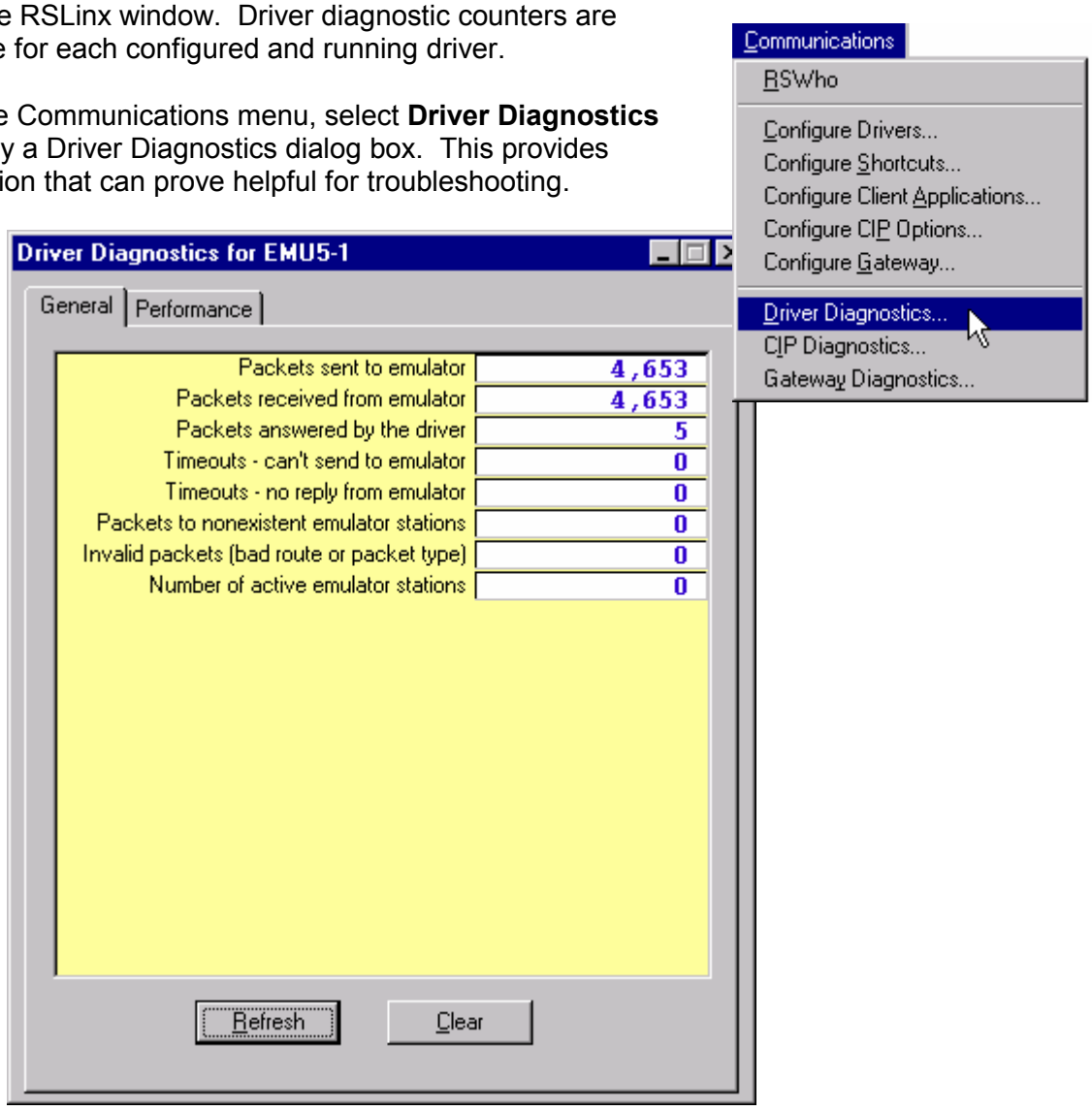


Training Section 12: Diagnostics, Troubleshooting and Options

Once a link is established, we can look back into RSLinx to see some of its diagnostic and troubleshooting features (this section assumes the link you established in the previous section is still active).

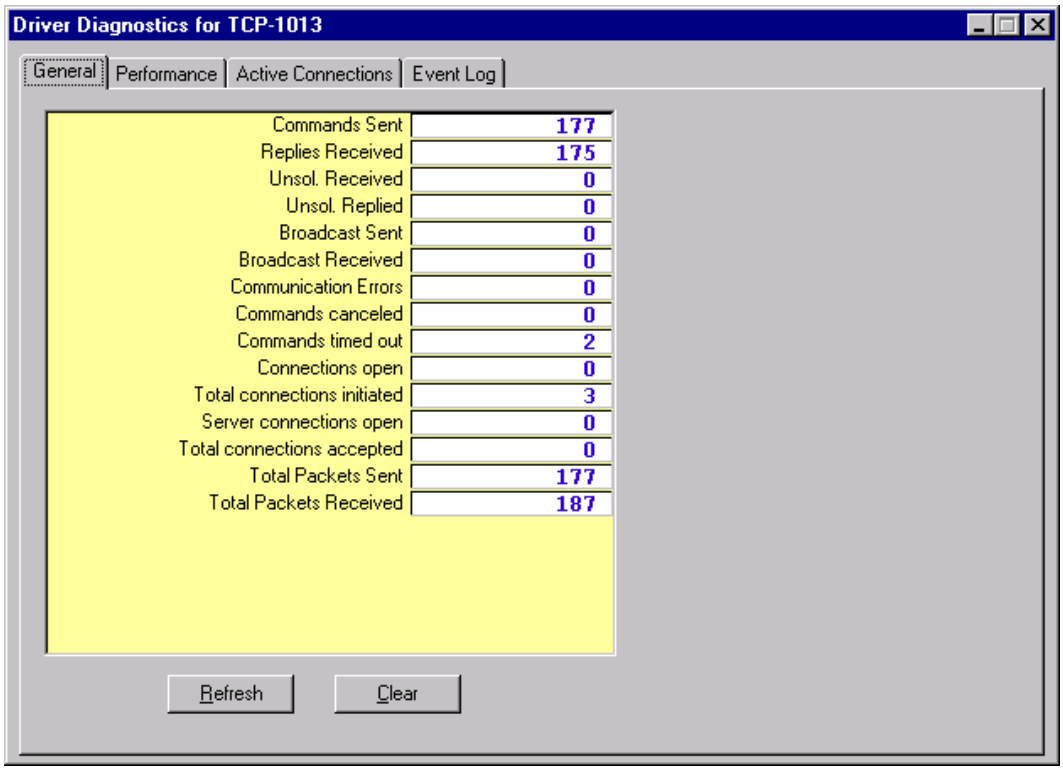
Open the RSLinx window. Driver diagnostic counters are available for each configured and running driver.

From the Communications menu, select **Driver Diagnostics** to display a Driver Diagnostics dialog box. This provides information that can prove helpful for troubleshooting.



More TCP diagnostics are found under the General, Performance, Connection Event Log, and Active Connections tabs in the tabbed dialog.

TCP diagnostics are modeless windows; therefore, they can remain on the desktop if you minimize RSLinx. The **Driver Diagnostics for TCP** box is shown below:



Server Diagnostics are found under the DDE/OPC menu - Server Diagnostics. Server diagnostics are modeless; therefore, they can remain on the desktop if you minimize RSLinx.

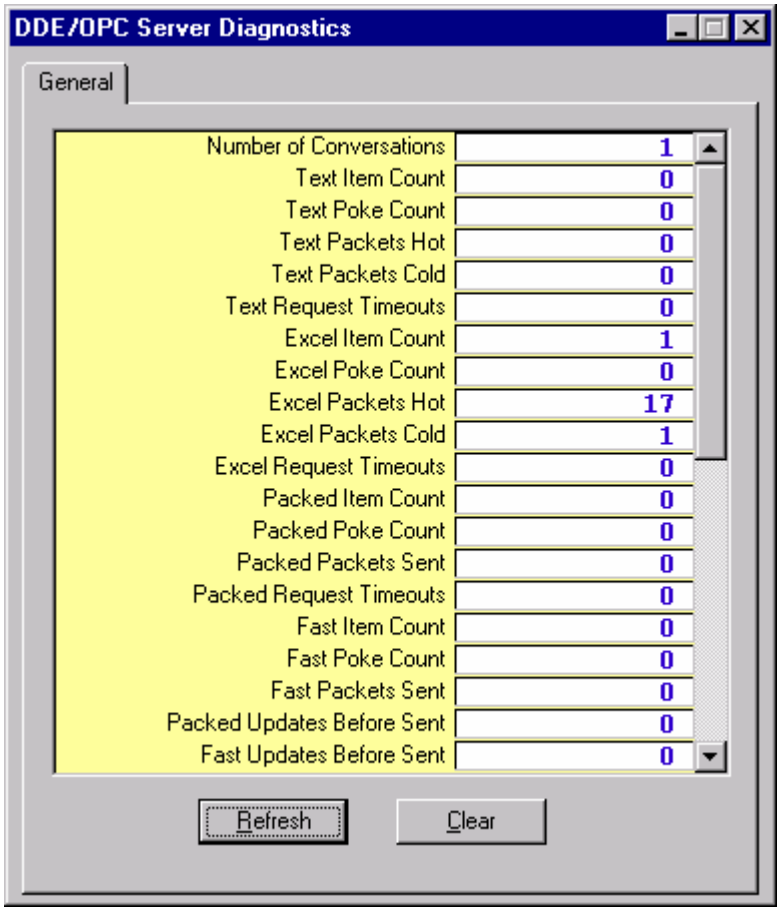
Select Server Diagnostics from the DDE/OPC menu to display the following diagnostic counters:

- Number of Conversations - The number of DDE Conversations handled by RSLinx. This is normally the same as the number of clients talking to RSLinx; however, this is not always the case.
  - Using a Visual Basic client, each DDE display is another conversation to RSLinx. Therefore, you could have a Visual Basic Client requesting five separate items, and have five conversations, or have Excel requesting thousands of items, and have only one conversation.
- Text Item Count - The number of DDE Items talking to RSLinx via the CF_TEXT DDE Protocol.
- Text Poke Count - The number of DDE Pokes sent to RSLinx from DDE Clients via the CF_TEXT DDE Protocol.

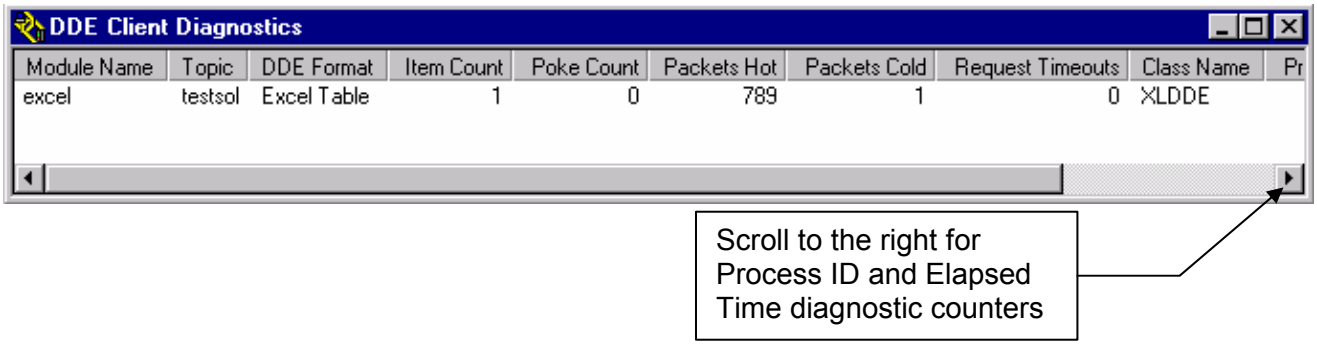
- Text Packets Hot - The number of DDE Advises sent from RSLinx to DDE Clients via the CF_TEXT DDE Protocol.
- Text Packets Cold - The number of DDE Requests sent from RSLinx to DDE Clients via the CF_TEXT DDE Protocol.
- Text Request Timeouts - The number of timeouts on a DDE Client's request of items via the CF_TEXT DDE Protocol.
- Excel Item Count - The number of DDE Items talking to RSLinx via the XL_Table DDE Protocol.
- Excel Poke Count - The number of DDE Pokes sent to RSLinx from DDE Clients via the XL_Table DDE Protocol.
- Excel Packets Hot - The number of DDE Advises sent from RSLinx to DDE Clients via the XL_Table DDE Protocol.
- Excel Packets Cold - The number of DDE Requests sent from RSLinx to DDE Clients via the XL_Table DDE Protocol.
- Excel Request Timeouts - The number of timeouts on a DDE Client's request of items, via the XL_Table DDE Protocol.
- Packed Item Count - The number of DDE Items talking to RSLinx via the PackedDDE DDE Protocol.
- Packed Poke Count - The number of DDE Pokes sent to RSLinx from DDE Clients via the PackedDDE DDE Protocol.
- Packed Packets Sent - The number of DDE updates sent from RSLinx to DDE Clients via the PackedDDE DDE Protocol.
- Packed Request Timeouts - The number of timeouts on a DDE Client's request of items via the PackedDDE DDE Protocol.
- Fast Item Count - The number of DDE Items talking to RSLinx via the FastDDE DDE Protocol.
- Fast Poke Count - The number of DDE Pokes sent to RSLinx from DDE Clients via the FastDDE DDE Protocol.
- Fast Packets Sent - The number of DDE updates sent from RSLinx to DDE Clients via the FastDDE DDE Protocol.

***Note**: Recall that "Packed" refers to the AdvanceDDE format and "Fast" refers to Wonderware's format.*

The DDE/OPC Server Diagnostics Dialog Box



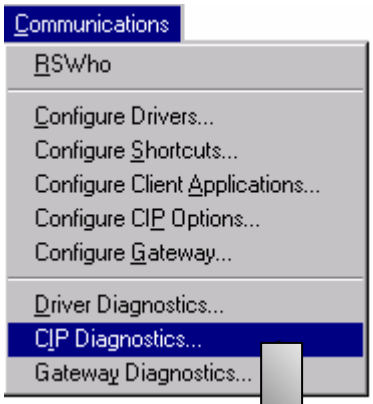
Select **DDE Client Diagnostics** from the DDE/OPC menu to display the following diagnostic counters:



Scroll to the right for Process ID and Elapsed Time diagnostic counters

- Module Name -The application's executable name that is talking to RSLinx.
- Topic - The DDE Topic the DDE Client is talking to.
- DDE Format - The DDE Format the DDE Client is using in this topic.
- Item Count - The number of items on this conversation in this DDE Format.
- Poke Count - The number of pokes sent via the DDE Client module in this DDE Format.
- Packets Hot - The number of advise updates sent to the DDE Client module in this DDE Format.
- Packets Cold - The number of request updates sent to the DDE Client module in this DDE Format. A conversation in PackedDDE or FastDDE does not support requests, so this is always zero for those formats.
- Request Timeouts - The number of requests that failed during a request for data in this DDE Format.
- Class Name - The class name of the module's DDE handling window. This is useful to determine whether the client is using DDEML or using DDE Messages.
- Process Id - The process ID of the application. It is useful to determine if there are two or more instances of the above module name running and whether it is a separate conversation from the program, or whether it is another instance.
- Elapsed Time - The length of time this conversation has been connected to RSLinx.

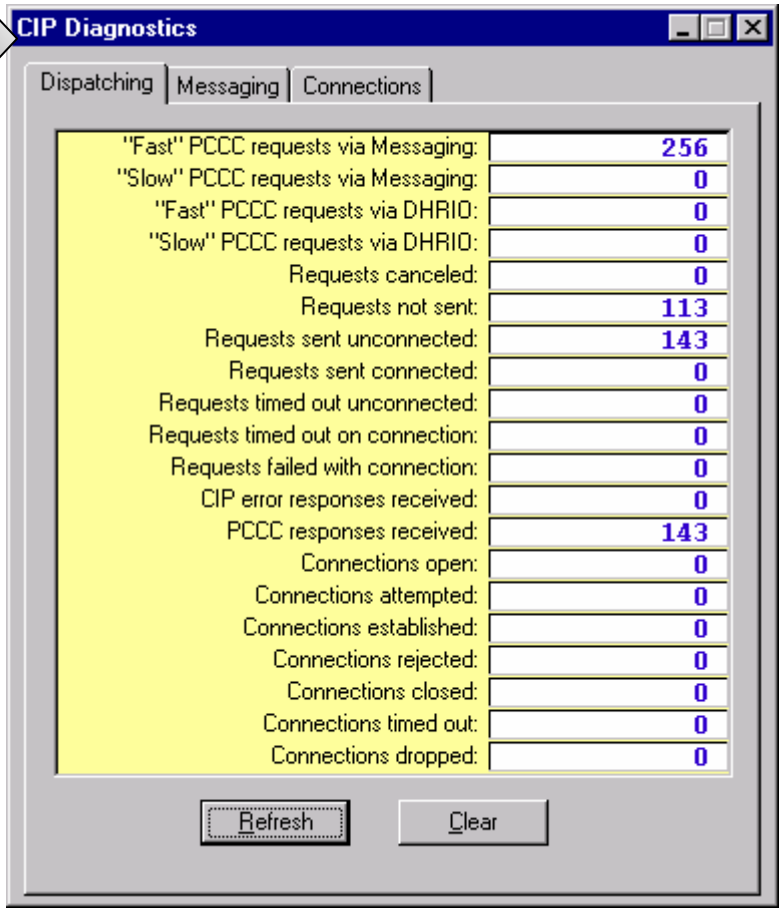
The CIP Diagnostics Dialog Box



The CIP Diagnostics dialog provides global information that is not specific to any driver.

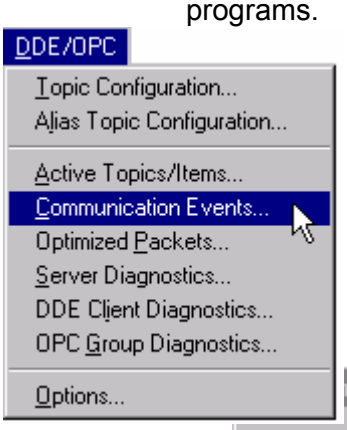
The CIP Diagnostics dialog currently displays CIP Dispatching, CIP Messaging, and CIP Connections diagnostics. Diagnostic counters shown are only available in this dialog; they are not available via RSWho.

For additional help on this topic, refer to the help FIND tab and type “CIP” (no quotes).

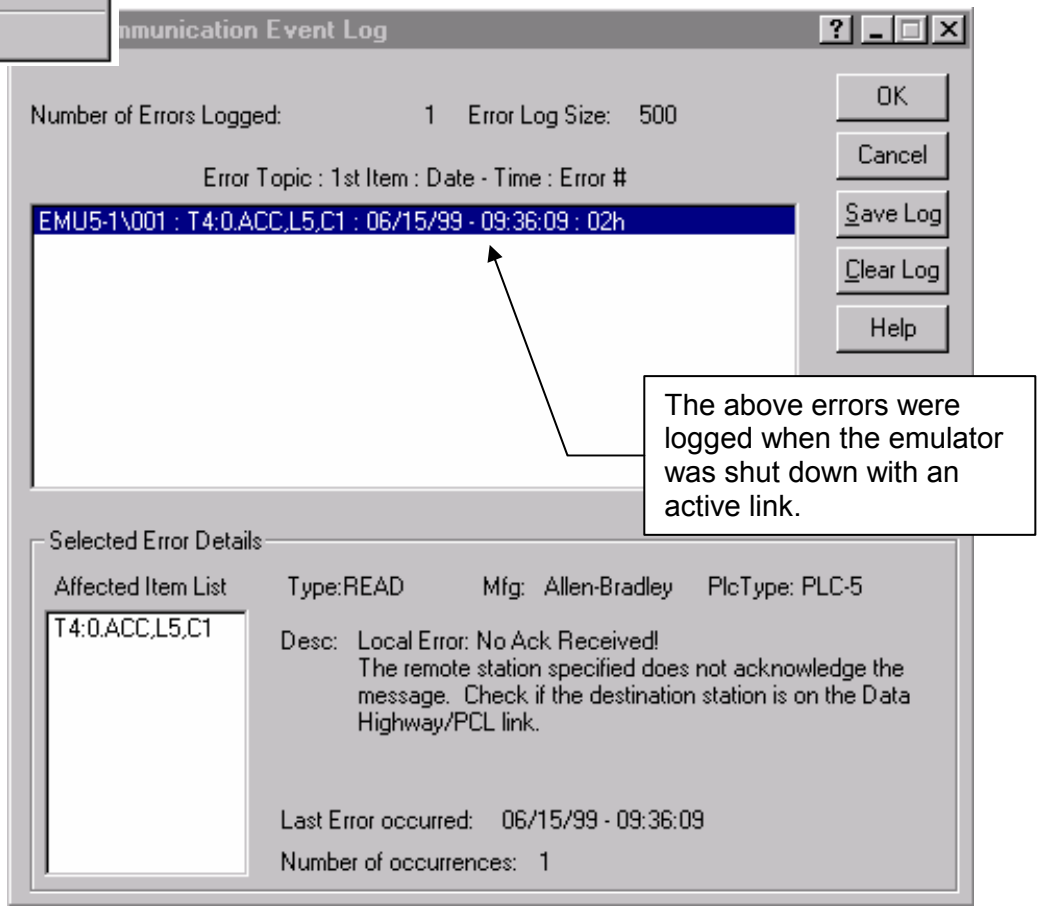


The DDE Communication Event Log

Communication Events from the DDE/OPC menu displays information on any DDE error messages logged while running RSLinx with DDE compliant programs.

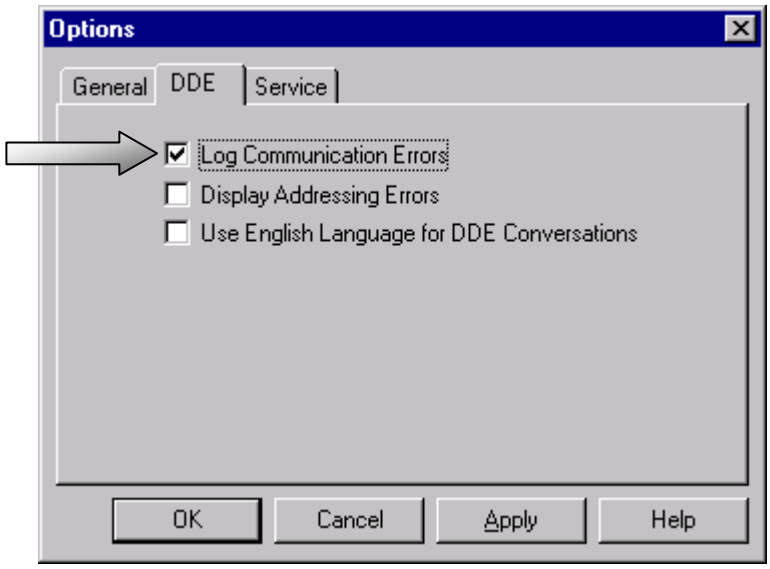


Each time a DDE error message is seen by RSLinx, the number of errors logged counter is increased by one and the entry is added to the window viewer. For a more detailed look at the information concerning the logged error, click the message in the window viewer and look at the information added to the Selected Error Details section near the bottom of the dialog box.



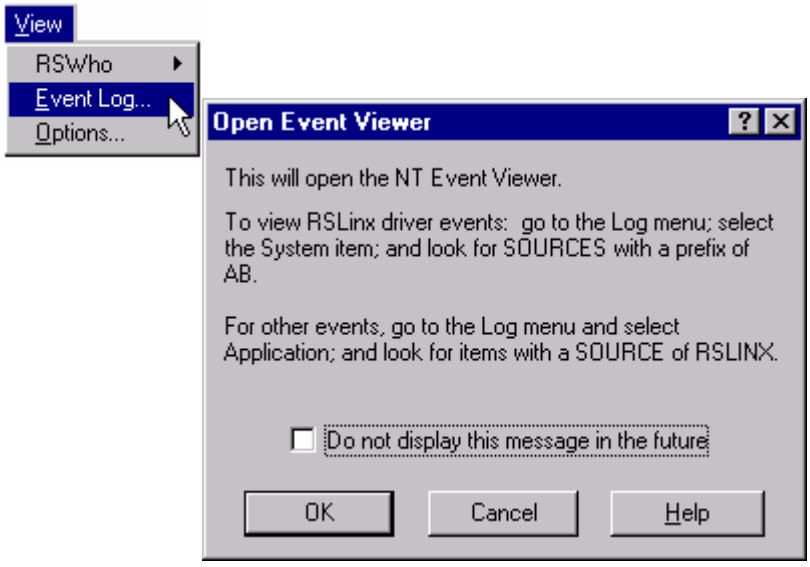
Communication events displays details such as processor manufacturer and type, the item affected by the error, and the DDE command type of the message.

- To enable logging of DDE errors, select Options from the DDE/OPC menu, and select the Log Communication Errors checkbox.

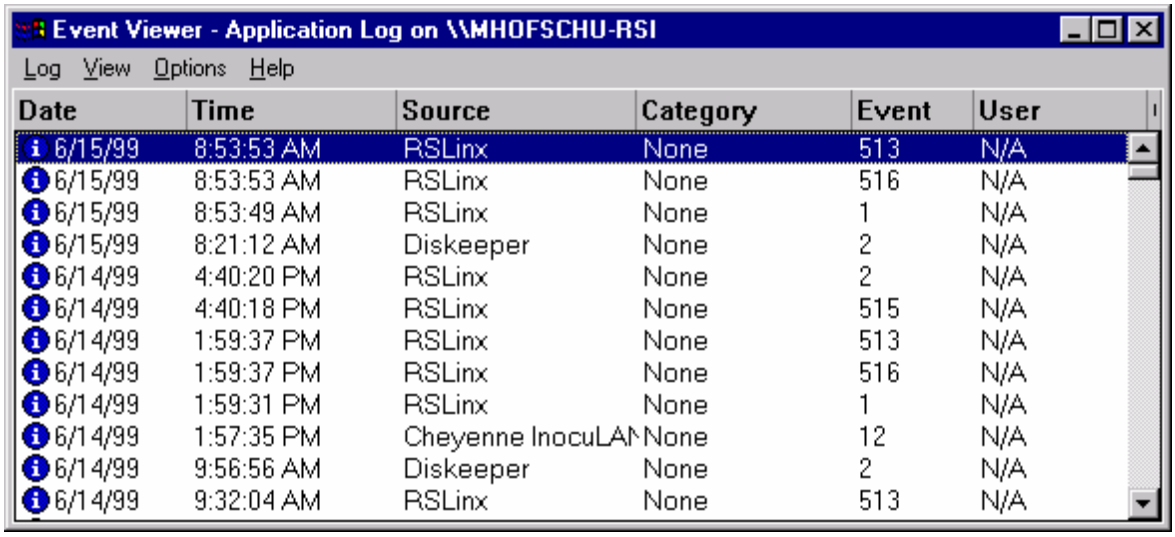


The Event Log

When you select View Event Log from the Tools menu, an Open Event Viewer dialog box displays:



The NT Event Viewer records important system occurrences. For RSLinx, events such as drivers starting and stopping successfully are displayed.

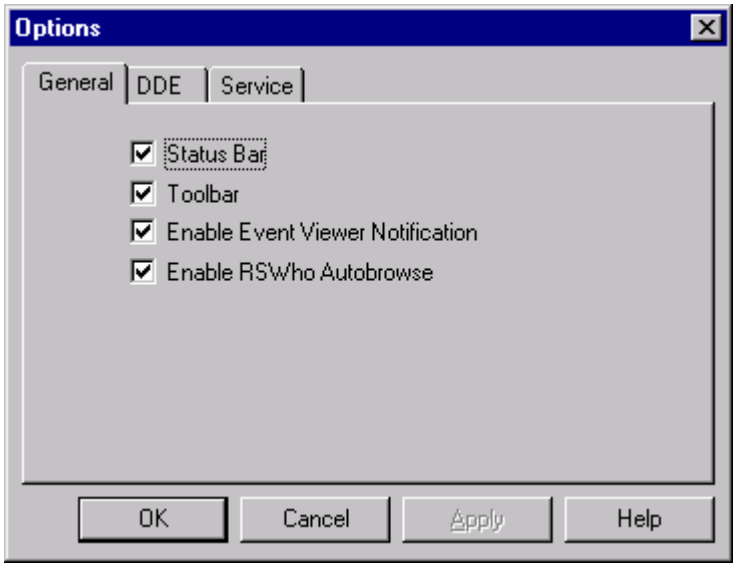


Date	Time	Source	Category	Event	User
6/15/99	8:53:53 AM	RSLinx	None	513	N/A
6/15/99	8:53:53 AM	RSLinx	None	516	N/A
6/15/99	8:53:49 AM	RSLinx	None	1	N/A
6/15/99	8:21:12 AM	Diskeeper	None	2	N/A
6/14/99	4:40:20 PM	RSLinx	None	2	N/A
6/14/99	4:40:18 PM	RSLinx	None	515	N/A
6/14/99	1:59:37 PM	RSLinx	None	513	N/A
6/14/99	1:59:37 PM	RSLinx	None	516	N/A
6/14/99	1:59:31 PM	RSLinx	None	1	N/A
6/14/99	1:57:35 PM	Cheyenne InocuLAN	None	12	N/A
6/14/99	9:56:56 AM	Diskeeper	None	2	N/A
6/14/99	9:32:04 AM	RSLinx	None	513	N/A

For more information on the Event Viewer, search the help that accompanies the Event Viewer.

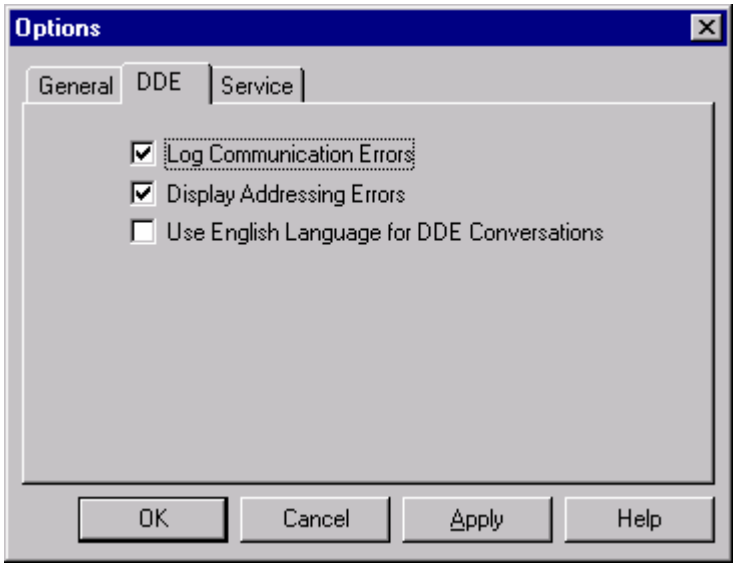
Options
When you select Options from either the DDE/OPC or Tools menu, an Options dialog box displays. The Options dialog box contains three tabs:

The General Tab allows you to toggle on and off RSLinx screen viewing options such as the Status Bar and Tool Bar. It also enables or disables two options: Event Viewer notification and the RSWho autobrowse feature.

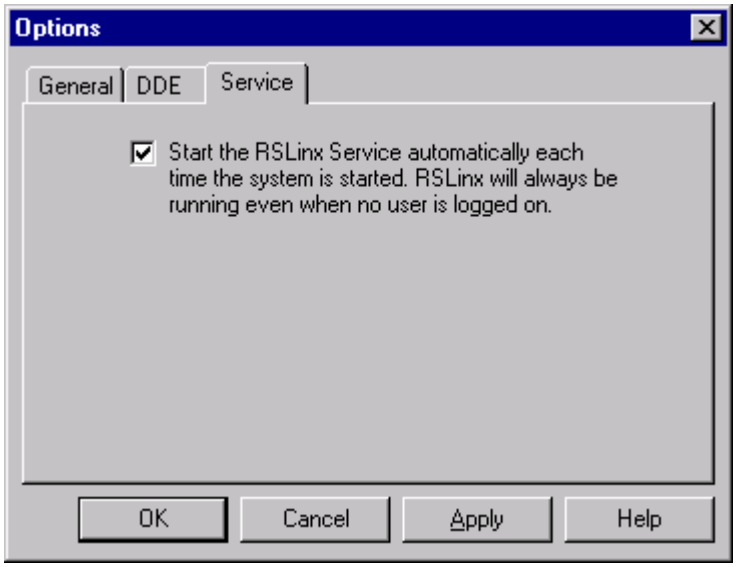


The DDE Tab provides the ability to log communication errors and display addressing errors.

- *Use English Language for DDE Conversations option*: provides the option of using English conventions for numbers when converting DDE data in foreign languages.
- CF_TEXT based floating point uses English floating point separators with this selected, and the language specific separators if not selected.
- If the machine's language is changed, RSLinx will use the new language unless this is selected.



The Services Tab gives the option of starting RSLinx as an NT Service each time your system is started. If you are using RSLinx as a server, it must be running as an NT service.



It is usually recommended to run RSLinx as a service on NT and Remote OPC works best when RSLinx is run as a service.

Running RSLinx as a service requires no user logon, and user logoff will NOT automatically close RSLinx. Remember, though, to stop the RSLinx service when moving ANY Rockwell Software activation

- See note below on how to do it…

Important: *When running RSLinx as a service on an NT operating system, do not open it as an application! It can be opened and shut down by right clicking the icon in the taskbar system tray (far lower right of screen).*

Appendix A: Support Library Technotes

The following technotes and more can be found in the Support Library at the Rockwell Software website. Check here first to answer most of your questions and discover the latest about any Rockwell Software product:

http://support.software.rockwell.com/supportlibrary/

http://software.rockwell.com/

http://suppport/rockwellautomation.com/

Rockwell Software
Support Library

NetDDE NT to NT

The information in this technote applies to:

Product:	**RSLinx**	Type:	**Application Note**
		Technote ID:	**A958**
Package/Module(s):	**RSLinx for NT**	Modified:	**03/29/99**
Revision(s):	**N/A**		
Fixed In Revision:	**N/A**		
Sub-System(s):	**DDE Server**		

Technical Note Details:

Syntax in Excel when doing NetDDE from an NT machine to another NT machine:
='\\ComputerName\NDDE$'|'DDEShareName'!'N7:0'

When doing NetDDE from an NT machine to another NT machine on the **same Domain** then no password is needed.

When Doing NetDDE from an NT machine to another NT machine on a **different Domain** then Excel prompts the user for a password (need to enter the password of the NT server box).

Legal Notice:

Rockwell Software
Support Library

NetDDE Configuration with RSLinx, RSView32, and EXCEL

The information in this technote applies to:

Product:	**RSView32**	Type:	**Application Note**
		Technote ID:	**Q528**
Package/Module(s):	**Works**	Modified:	**02/16/99**
Revision(s):	**5.01.23**		
Fixed In Revision:			
Sub-System(s):	**Dynamic Data Exchange**		

Technical Note Details:

Background:
This application note covers the following NetDDE examples:

RSView32 connecting to RSLinx
RSView32 connecting to RSView32
EXCEL 97 connecting to RSView32
NetDDE common pitfalls

RSView32 connecting to RSLinx:

1. Configure RSLinx to communicate to the PLC.
2. Create a DDEshare name according to the RSLinx NetDDE help.
3. If you are using a Windows 95 box as the client run NetDDE.
4. Create a DDE node in RSView32 with the following information:
   Application = \\computer_name\ndde$
   Topic = dde_share_name
5. Create an RSView32 database tag using the DDE node and a PLC address.
   address = s:23

RSView32 connecting to RSView32:

1. Start the DDE server in RSView32. To do this check the DDE Server check box in the RSView32 startup editor, or execute the command DDEServerOn.
2. Create a DDE node in RSView32 with the following information:
   Application = \\computer_name\ndde$
   Topic = RTDataProject_Name$
   The RSView32 DDEshare is automatically configured as RTDataProject_Name$ This DDEshare name must be used in the RSView32 client topic configuration.
3. Set the RSView32 client tag address to the name of the tag on the RSView32 Server.

EXCEL connecting to RSView32:

1. Start the DDE server in RSView32. To do this check the DDE Server check box in the RSView32 startup editor, or execute the command DDEServerOn.
2. Client application name = '\\computer_name\ndde$'
3. The RSView32 DDEshare is automatically configured as RTDataProject_Name$ This DDEshare name must be used in the EXCEL topic configuration.
4. The EXCEL item should be the name of the RSView32 Server tag.
5. The complete entry in the EXCEL 97 cell would appear as: ='\\computer_name\ndde$'|'RTDataProject_Name$'!'tag_name'

NetDDE Common Pitfalls:

- The server node name cannot have a dash "-" in the name. Use an underscore "_" instead of a dash.
- NetDDE is not running on a Windows95 client. To start NetDDE click on start, run and run "NetDDE".
- #N/A means that the link could not be established which may indicate the user logging in does not have read rights to the data. If "everyone" in the DDEshare permissions has rights try creating a specific user and give that user read rights. If there is a user name created try deleting and re-create the user. A user may exist but not be functional.
- In DDEshare the application and topic name was not under the **STATIC** section.
- In DDEshare under trusts the start application enabled and initiate to application enabled must be checked.
- The DDEPokeEnable command must be issued at the RSView32 server to enable DDE poking of data from a client.

Legal Notice:

The information provided in the Rockwell Software Inc. (RSI) Knowledgebase is provided "as is" without warranty of any kind. RSI disclaims all warranties, either expressed or implied and in no event shall RSI be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of profit or damage, even if RSI have been advised of the possibility of such damages.

The foregoing limitation may not apply in those states that do not allow the exclusion or limitation of liability for damages.

Rockwell Software
Support Library

The OPC client poll rate can over ride the RSLinx topic poll rate

The information in this technote applies to:

Product:	**RSLinx**	Type:	**Release Note**
		Technote ID:	**R221**
Package/Module(s):	**RSLinx (All)**	Modified:	**12/28/98**
Revision(s):	**2.x**		
Fixed In Revision:	**N/A**		
Sub-System(s):	**DDE/OPC Server**		

Technical Note Details:

If the update rate specified by an OPC client is faster than the poll rate specified in RSLinx. RSLinx will request data from the PLC as the update rate specified by the OPC client

Example: If the poll rate in RSLinx is set to 1000ms, and the update rate of the OPC client is set for 200ms then RSLinx will request data from the PLC every 200ms.

Example: If the poll rate in RSLinx is set to 200ms, and the update rate of the OPC client is set for 1000ms then RSLinx will request data from the PLC every 200ms. However, the OPC client will only request the data from RSLinx every 1000ms

Note: DDE links will update at the actual poll rate RSLinx uses. In the case where the OPC client update rate is faster than the RSLinx poll rate, DDE clients will also be updated at the rate of the OPC client

Legal Notice:

Rockwell Software
Support Library

TEN THINGS YOU SHOULD KNOW ABOUT TCP/IP

The information in this technote applies to:

Product:	**RSLinx**	Type:	**Application Note**
		Technote ID:	**A1152**
Package/Module(s):	**RSLinx (All)**	Modified:	**03/29/99**
Revision(s):	**N/A**		
Fixed In Revision:	**N/A**		
Sub-System(s):	**N/A**		

Technical Note Details:

TEN THINGS YOU SHOULD KNOW ABOUT TCP/IP

1) What is TCP/IP?

TCP/IP is a suite of protocols that can be used to route information. TCP/IP stands for Transmission Control Protocol/Internet Protocol. This suite of protocols can provide addressing in wide-area networks (WANs) and can provide connectivity to a variety of hosts. TCP/IP is the protocol that the Internet is built on.

2) What is TCP/IP Addressing?

A TCP/IP address is numeric identifier assigned to each host on the network. Assigning a TCP/IP address allows the software assigned TCP/IP address to be mapped to the hardware encoded MAC address of the network interface card (NIC).

3) How is the TCP/IP Address formatted?

A TCP/IP address is comprised of 32-bits of information. These 32-bits are then divided into four 8-bit sections. These sections are referred to as octets. The octets are typically displayed in decimal format (i.e. 131.150.186.10). This type of format is commonly referred to as dotted decimal notation. Octets can also be displayed in binary format (i.e. 10000011.10010110.10111010.00001010).

4) What are the Network Address and the Node Address?

To allow for efficient routing over the Internet, the TCP/IP address format was broken into two parts. The first part is the Network Address. The Network Address is used to uniquely identify each network. Every host in the same network shares the same Network Address. The second part is the Node Address. The Node Address uniquely identifies each host on the network. The Node Address is commonly referred to as the Host Address.

5) What are Network Classes?

Network Classes are used to help determine how to subdivide the TCP/IP address into the Network Address and the Node Address. The three most common Network Classes are Class A, Class B, and Class C. The Network Classes were created based on the actual network size. Depending on the number of required hosts in the network, this determines which Network Class will be assigned.

6) How do I know which Network Class I am using?

In order to ensure efficient routing between different Network Classes, only the leading bit pattern of the first octet is used to distinguish between the different Network Classes. To decipher the leading bit pattern of the first octet, you must convert the octet from decimal to binary format.

The Class A network has the first octet with the leading bit pattern 0.
The Class B network has the first octet with the leading bit pattern 10.
The Class C network has the first octet with the leading bit pattern 110.

7) What are the specifications for a Class A, Class B, and Class C network?

In a Standard Class A network, the first octet is reserved for the Network Address, and the remaining three octets are reserved for the Node Addresses. The Class A format is:
Network.Node.Node.Node
To determine the valid number of Network Addresses, recall that a Class A network has a first octet with the leading bit pattern 0. This leaves only 7-bits available, from the first octet, for Network Addresses. Converting this binary number to decimal gives a total of 128 (0-127 decimal) possible Network Addresses. A Network Address of 0 is invalid and cannot be used. Also, a Network Address of 127 is invalid and cannot be used. Thus, the only valid Class A Network Addresses available will be between 1-126 decimal. This means that any TCP/IP address that begins with a number between 1 and 126 is a Class A Network.
To determine the valid number of Node Addresses, use the remaining three octets for addressing the hosts. Converting this 24-bit binary number to decimal gives a total of 16,777,216 possible Node Addresses. A Node Address of all zeros and all ones is invalid. Thus, this leaves a total of 16,777,214 possible Node Addresses available for a Standard Class A Network.

In a Standard Class B network, the first two octets are reserved for the Network Address, while the next two octets are reserved for the Node Addresses. The Class B format is:

Network.Network.Node.Node

To determine the valid number of Network Addresses, recall that a Class B network has a first octet with the leading bit pattern 10. This leaves only 14-bits available, from the first two octets, for Network Addresses. Converting this binary number to decimal gives a total of 16,384 possible Network Addresses. Thus, the only valid Class B Network Addresses available will always have a first octet that begins with a decimal number between 128-191 decimal. This means that any TCP/IP address that begins with a number between 128 and 191 is a Class B Network.

To determine the valid number of Node Addresses, use the remaining two octets for addressing the hosts. Converting this 16-bit binary number to decimal gives a total of 65,536 possible Node Addresses. A Node Address of all zeros and all ones is invalid. Thus, this leaves a total of 65,534 possible Node Addresses available for a Standard Class B Network.

In a Standard Class C network, the first three octets are reserved for the Network Address, while the last octet is reserved for the Node Addresses. The Class C format is:

Network.Network.Network.Node

To determine the valid number of Network Addresses, recall that a Class C network has a first octet with the leading bit pattern 110. This leaves only 21-bits available, from the first three octets, for Network Addresses. Converting this binary number to decimal gives a total of 2,097,152 possible Network Addresses. Thus, the only valid Class C Network Addresses available will always have a first octet that begins with a decimal number between 192-223 decimal. This means that any TCP/IP address that begins with a number between 192 and 223 is a Class C Network.

To determine the valid number of Node Addresses, use the last octet for addressing the hosts. Converting this 8-bit binary number to decimal gives a total of 256 possible Node Addresses. A Node Address of all zeros and all ones is invalid. Thus, this leaves a total of 254 possible Node Addresses available for a Standard Class C Network.

Class Decimal Range of First Octet Maximum Networks Maximum Hosts
A 1-126 126 16,777,214
B 128-191 16,384 65,534
C 192-223 2,097,152 254

8) What is a Subnet Mask?

A Subnet Mask is a 32-bit value that allows the TCP/IP protocol stack to distinguish the Network Address from the Node Address.

9) What are valid Subnet Masks for Standard Class A, Class B, and Class C networks?

Standard Addressing by definition means that no Subnetting is being used on the network. This means that the entire Node Address must be reserved for hosts.

Standard Class A = 255.0.0.0
Standard Class B = 255.255.0.0
Standard Class C = 255.255.255.0

10) Are there any invalid TCP/IP Addresses?
Invalid Addresses Examples
- TCP/IP address of all ones IP Address = 255.255.255.255
- TCP/IP address of all zeros IP Address = 0.0.0.0
- Network Address of all ones IP Address = 255.255.255.100, Subnet Mask = 255.255.255.0
- Network Address of all zeros IP Address = 0.0.0.100, Subnet Mask = 255.255.255.0
- Network Address of 127 IP Address = 127.100.100.100, Subnet Mask = 255.0.0.0
- Node Address of all ones IP Address = 100.255.255.255, Subnet Mask = 255.0.0.0
- Node Address of all zeros IP Address = 100.0.0.0, Subnet Mask = 255.0.0.0

Rockwell Software
Support Library

How to monitor which topic within an Alias is currently active

The information in this technote applies to:

Product:	**RSLinx**	Type:	**Application Note**
		Technote ID:	**A1143**
Package/Module(s):	**RSLinx (All)**	Modified:	**03/01/99**
Revision(s):	**N/A**		
Fixed In Revision:	**N/A**		
Sub-System(s):	**DDE Server**		

Technical Note Details:

Background:

RSLinx's "Alias" topic allows users to specify alternate communications paths. For example, a user could configure an alias called "MyAlias" consisting of many topics. If the first topic stops communicating, the Alias can automatically switch to the next topic as needed. This switchover is completely transparent to the application, which only sees the alias topic.

In some applications, a user may wish to indicate which topic within the alias is the currently active topic. This can allow an application such as RSView to display an indicator to an operator that a problem exists. It can also allow an application to take some sort of programmed action when a communications failure takes place.

Solution:

RSLinx provides an indicator for each alias topic called "ActiveTopic". The following DDE link will return the name of the active alias topic:

RSLinx!AliasTopicName|'Activetopic'

Example:

1. In RSLinx, configure an topic called "MyAlias". Within the alias configure topics, configure topics "Topic1" through "Topic4". In Microsoft Excel, enter the following formula:

=RSLinx|'MyAlias'!'Activetopic'

2. In order to control which topic is active, enter the following code into the Microsoft VBA editor:

```
Public Sub SetMyActiveTopic()
'Dim variables for DDE channel and Active Topic
Dim usrChan As Long
Dim usrActiveTopic As String

'Open a channel to RSLinx
usrChan = DDEInitiate("RSLinx", "System")

'Open an input box to allow the operator to enter the desired topic
usrActiveTopic = InputBox("Enter Active Topic: ", "Topic Selection")

'Set the active topic based on the input box
Application.DDEExecute usrChan, "[Set_Alias(MyAlias," &
usrActiveTopic & ")]"

'Close the channel
Application.DDETerminate usrChan
End Sub
```

3. Finally, create a DDE link in one of Microsoft Excel's cells.

=RSLinx|'MyAlias'!'address'

where 'address' is a valid address in the PLC. Now run the VBA script from the VBA editor. When the input box appears, enter the name of one of the topics within MyAlias. When the script has finished running, the name of the topic that was entered should appear in step 1.

For more help on Alias topics, see the RSLinx help file.

Rockwell Software
Support Library

Understanding How RSView32 Manages Communications With RSLinx - or Any DDE/OPC Server.

The information in this technote applies to:

Product:	**RSView32**	Type:	**Application Note**
		Technote ID:	**A951**
Package/Module(s):	**Runtime Works**	Modified:	**12/15/98**
Revision(s):	**6.00.41**		
Fixed In Revision:			
Sub-System(s):	**RSLinx**		

Technical Note Details:

Background:

RSView32 has implemented a very sophisticated mechanism for only requesting the data that is needed from the data server. This means that as screens change in the HMI, RSView32 asks the data server for different data. When RSView32 is asking RSLinx for DDE or OPC data, then the Active Items list can be viewed to determine exactly what RSView32 is asking RSLinx for. When RSView32 is asking RSLinx for Direct or C-API data, then the Active Items list does not show the data being requested.

Understanding how RSView32 asks the data server for data is important when optimizing communications, and when implementing unsolicited message data, which is data that originates from the PLC, or data not being polled by the HMI.

Show Last Acquired Value in Graphics:
By default graphic screens have this property checked or enabled. The last acquired value is displayed on a graphic, rather than a wireframe outline indicating the data is not fresh, when this property is enabled. The last acquired value is never displayed when the tag is in Error.
For more information on Tag states:
Jump to A202 - Tag State Information

This property is intended to show the user the last POLLED value, for data that is slow to update, such as a dialup modem network. Remember when the screen is closed all data important to that screen is no longer needed, and is removed from the Active Items list in RSLinx. The next time the screen is displayed, the data is initially read. The Show Last Acquired Value screen setting is intended to present the last value read while waiting for the new value to be read.

If an unsolicited write comes in while the screen is closed, then RSLinx has no place to send the data, so it is lost. RSLinx does not buffer this data. The next time the screen is displayed the data is stale or uninitialized and appears to be wire framed on the graphic display regardless of the setting of Show Last Acquired Value. If however, the topic in RSLinx has been configured to be polled as well as accept unsolicited data, then each time the screen is displayed, the data will be read. The object remains wireframed until the read is complete.
For more information on how to successfully implement unsolicited / polled data in the HMI:
Jump to A449 Receiving Unsolicited Messages in RSView32 with RSLinx

One exception to this active items list rule is when the Display Setting Cache After Displaying is set to Yes, and the Always Updating option is checked. This forces all screen data to be continually polled. Other exceptions are when the data is also being used in a background task such as Datalog, Alarming, Events, Derived tags or a tag object has been Dim'ed WithEvents in VB or VBA. Whenever a task requires the data then it remains on the Active Items list in RSLinx even when the graphic is closed.

Once RSView32 accesses data associated with a topic, that topic becomes locked in RSLinx. A locked topic can't be modified, except the poll rate can be changed. The topic remains locked even after the graphic is closed, or the project is stopped. To unlock a topic the project must be closed. Topics remain locked after a graphic screen is closed or the project is stopped for performance optimization.

Communication Status Information:

In RSLinx when using DDE/OPC there are diagnostic tools for troubleshooting communications problems. The Active Items List shows the addresses currently being requested by RSView32. The Communication Event Log shows the last 500 errors, and the affected item in the list. The Optimized Packets shows how RSLinx is optimizing the data requested by RSView32, and shows the number of packets received, how many are valid, and how many are invalid. Client and Server Diagnostics can be referenced for additional information.

In RSView32, there exists some communication status tags that can be referenced for trouble shooting communications problems. These tags are only updated when an application is referencing them. So in other words, if the tags are read after a communication error they are uninitialized. For this reason, it is a good idea to create a datalog model - which only logs on demand - for these system tags. This will keep the tags current and fresh by continually referencing them. A list of these tags follows...

When channels are configured in RSView32, the following 4 system tags are created for each channel. Here channel 1 has been configured for DH+.

System\ComErrorString1 - the last error that occurred
System\ComErrorValue1 - the last error number that occurred
System\ComStatusString1 - the current status, usually Success - no error.
System\ComStatusValue1 - the current status number, 0 = success no error

When a DDE node is created in RSView32, the following 4 system tags are created.

System\ComErrorStringDDE
System\ComErrorValueDDE
System\ComStatusStringDDE
System\ComStatusValueDDE

When a OPC node is created in RSView32, the following 4 system tags are created.

System\ComErrorStringOPC
System\ComErrorValueOPC
System\ComStatusStringOPC
System\ComStatusValueOPC

All system communications tags may be viewed in Tag Monitor by entering System\Com* as the tag name, and selecting all.

Legal Notice:

Rockwell Software
Support Library

How to connect a remote RSView32 client to an RSLinx Gateway OPC Server.

The information in this technote applies to:

Product:	**RSView32**	Type:	**Application Note**
		Technote ID:	**A945**
Package/Module(s):	**Runtime Works 6.00.41**	Modified:	**06/03/99**
Revision(s):	**6.10.15 6.10.16**		
Fixed In Revision:			
Sub-System(s):	**OPC Comms RSLinx**		

Technical Note Details:

Background:
When connecting to RSLinx Gateway from a RSView32 client using Remote OPC, the user must perform some additional configuration steps when RSLinx is not installed on the client machine. This technote documents how to make the connection.

System Requirements:
To implement Remote OPC, there must be a Windows NT network with a Primary Domain Controller (PDC) that all computers login to. If the RSView32 client is running on Windows 95/98 DCOM must be installed and properly configured. For more information on Remote OPC system requirements consult the RSView32 online help, Requirements for using OPC on a Client/Server network.

RSLinx Gateway currently supports up to 5 remote clients. The easiest way to make a successful remote connection is to run RSLinx as a Service Automatically. For more information on DCOM configuration requirements for RSLinx,

Jump to A1239 DCOM Configuration Requirements for RSLinx Gateway Remote OPC

Enabling Client Software:
RSLinx Gateway CD provides the Remote OPC Client enabling software. This enabling software must be installed on each remote RSView32 OPC client where RSLinx is not installed.

Node Configuration:
For each RSLinx Gateway connection, a node must be configured in RSView32. In the Node configuration, select OPC Server and type a node name. Click the Server Name browse button and select the RSLinx OPC Server - Remote type (RSLinx 2.0), or RSLinx Remote OPC Server - Remote type (RSLinx 2.1). Click the Server Computer Name or Address browse button and browse to the remote OPC Server Computer.

The Access Path is optional in the node configuration. If specified, it is an OPC topic name in RSLinx on the remote OPC Server. When the Access Path is omitted from the node configuration, the address line of each tag must include the OPC topic name surrounded by brackets. For example, the address for a tag using N7:0 would be entered as [TopicName]N7:0.
The Update Rate defaults to 1 second.

This represents the frequency that the RSView32 OPC client will ask the RSLinx OPC server for an update. RSLinx Gateway also specifies a poll rate for each OPC topic. The default poll rate is 1000 ms or 1 second. When the OPC Topic poll rate is modified, then the Update Rate should be modified as well. The Update Rate should not be set faster than the corresponding OPC topic poll rate.

RSLinx Gateway Server name change:
RSLinx 2.0 name = RSLinx OPC Server
RSLinx 2.1 name = RSLinx Remote OPC Server.

Tip:
When upgrading from RSLinx Gateway 2.0 to 2.1 the node configuration in RSView32 should be altered to reflect the server name change. The old name remains configured as long as RSLinx 2.0 was installed.

Legal Notice:

Rockwell Software
Support Library

RSLinx OPC Server and RSView32

The information in this technote applies to:

Product:	**RSLinx**	Type:	**Application Note**
		Technote ID:	**A1181**
Package/Module(s):	**RSLinx (All)**	Modified:	**03/29/99**
Revision(s):	**2.1.xx**		
Fixed In Revision:	**N/A**		
Sub-System(s):	**OPC Server**		

Technical Note Details:

When using RSView and OPC to RSLinx a slight difference in configuration of the OPC connection is seen between RSLinx v2.00.97.30 and RSLinx v2.1.xx. The following refers to the RSView configuration to the RSLinx OPC server. The selections are found in the Node configuration dialogue under the Server section.

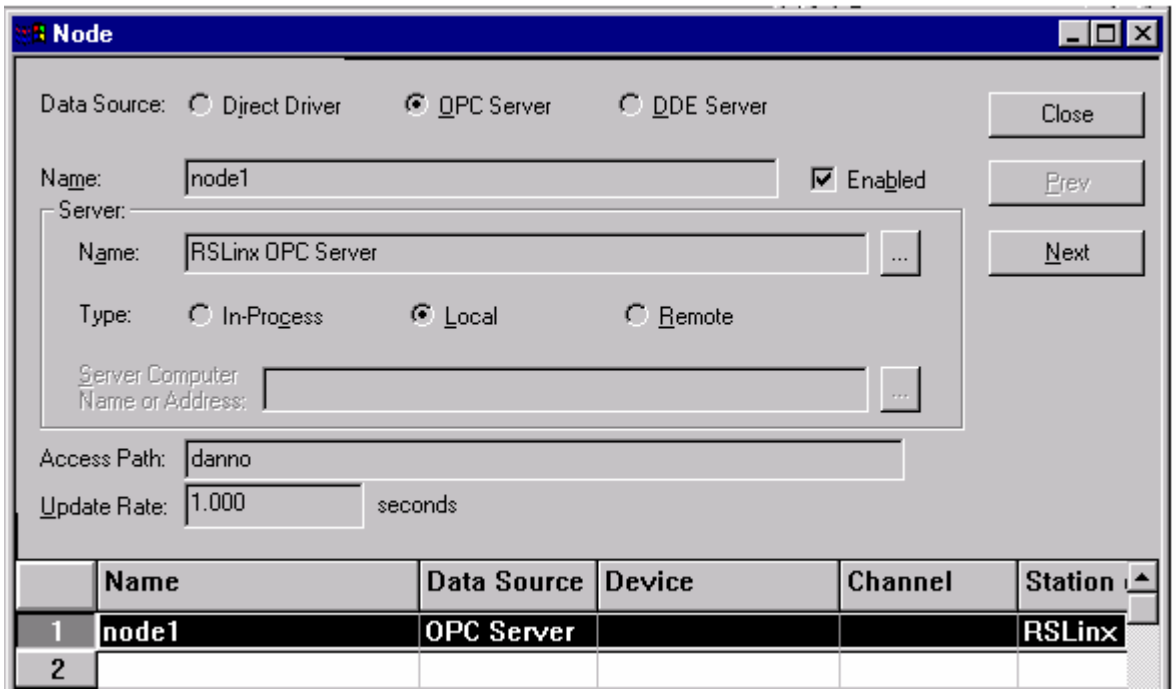
Using RSLinx 2.0 OPC server:

Local: The server name must be RSLinx OPC Server. The type must be Local.
Remote: The server name must be RSLinx OPC Server. The type will be Remote.

Using RSLinx v2.1.xx OPC server:

Local: The server name must be RSLinx OPC Server. The type must be In Process.
Remote: The server name must be RSLinx Remote OPC Server. The type must be Local.

This graphic for reference only.



Legal Notice:

Rockwell Software
Support Library

Client: Could not connect to the OPC Server - caused by Upgrading RSLinx Professional from 2.0 to 2.1

The information in this technote applies to:

Product:	**RSView32**	Type:	**Error Message**
		Technote ID:	**E945**
Package/Module(s):	**Runtime Works**	Modified:	**06/03/99**
Revision(s):	**6.10.16**		
Fixed In Revision:			
Sub-System(s):	**OPC Comms RSLinx**		

Technical Note Details:

Problem:
The following error is generated after upgrading RSLinx from 2.0 to 2.1 when OPC communications are in use between RSView32 and RSLinx Professional.

Client: Could not connect to the OPC Server.

Cause:
RSLinx Professional OPC Server version 2.1 now connects through a in-process DLL. RSLinx Professional OPC Server version 2.0 connected as a local EXE. This different connection method prevents RSView32 from communicating to the OPC server.

Solution:
Update the node definition, changing the RSLinx OPC Server Type from Local to In-Process.

Legal Notice:

Rockwell Software
Support Library

Cannot perform NetDDE with RSLinx on Windows 95/98

The information in this technote applies to:

Product:	**RSLinx**	Type:	**Problem / Solution**
		Technote ID:	**P687**
Package/Module(s):	**RSLinx for 95**	Modified:	**05/14/99**
Revision(s):	**N/A**		
Fixed In Revision:			
Sub-System(s):	**All DH**		

Technical Note Details:

Problem:
You will not be able to do NetDDE from one Windows 95 Computer to another Windows 95 Computer with RSLinx.

Cause:
RSLinx is looking for a DDESHARE Name in the Syntax that is needed for the Link item. Windows 95 does not have a DDESHARE Utility to create a DDESHARE. WINtelligent Linx created the DDESHARE when the DDE Topic was created.

Solution:
Preferred Solution - Use RSLinx Gateway and Remote OPC. This will require an OPC client such as RSView32 (v6.1x or later) or RSTools used in conjunction with RSJunctionBox for OPC (v1.6 or later of RSJunctionBox). There will also be an OPC AutomationDLL available as part of the RSLinx SDK package.

Alternate Solution - Run RSLinx on a Windows NT computer. This enables you to create the DDE Share that is necessary to do NetDDE.

Legal Notice:

Rockwell Software
Support Library

Checklist for trouble-free serial port communications.

The information in this technote applies to:

Product:	**Third Party**	Type:	**Application Note**
		Technote ID:	**A957**
Package/Module(s):	**Communications**	Modified:	**05/13/99**
Revision(s):	**N/A**		
Fixed In Revision:	**N/A**		
Sub-System(s):	**Serial Communications**		

Technical Note Details:

Below is a list of errors returned by various Rockwell Software products when attempting to configure a serial port communications driver.

"RSLinx has failed to configure the COM port." (RSLinx - AB_DF1)
"Failed to initialize PIC device driver." (RSLinx - AB_PIC)
"Device already open." (Wintelligent Linx)
"NETWORK_ERROR returned an error code of 7000." (A.I.-500)
"PCL_Link_Error returned an error code of 7000." (A.I.-5)
"Interface Hardware faulty or not present." (A.I.-2)
"SLC Response Timeout" (SLC Logistics)
"Serial port configuration error." (PCIS)

Here is a basic checklist to ensure trouble-free serial port communications.

DOS

Be sure the correct comm port is selected.
Be sure that the selected comm port actually exists on the computer.
(This information can be found by running MSD.EXE.)

Be sure that the comm port selected is enabled in the BIOS / CMOS setup.*
Be sure there are no conflicting devices on the selected comm port. (i.e. modems, Infrared ports, etc...)
Be sure there are no conflicting devices on the opposing comm port (ie. COM1 vs. COM3 and COM2 vs.COM4)
Be sure there are no low level drivers loading in the AUTOEXEC.BAT or CONFIG.SYS.

Windows 3.x

Be sure the correct comm port is selected.
Be sure that the selected comm port actually exists on the computer.
Be sure that the comm port selected is enabled in the BIOS / CMOS setup.*
Be sure there are no conflicting devices on the selected comm port. (i.e. modems, Infrared ports, etc...)
Be sure there are no conflicting devices on the opposing comm port (i.e. COM1 vs. COM3 and COM2 vs.COM4)
Be sure there are no low level drivers loading in the AUTOEXEC.BAT or CONFIG.SYS.
Be sure there are no drivers loading in the SYSTEM.INI ([386Enh] section), or in the WIN.INI (load= and run=)

Windows 95/98

Be sure the correct comm port is selected.
Be sure that the selected comm port actually exists on the computer.
Be sure that the comm port selected is enabled in the BIOS / CMOS setup.*
Be sure there are no conflicting devices on the selected comm port. (i.e. modems, Infrared ports, etc...)
(This information can be found in Device Manager)
Be sure there are no conflicting devices on the opposing comm port (i.e. COM1 vs. COM3 and COM2 vs.COM4)
(This information can be found in Device Manager)
Be sure there are no low level drivers loading in the AUTOEXEC.BAT or CONFIG.SYS.
Be sure there are no drivers loading in the SYSTEM.INI ([386Enh] section), or in the WIN.INI (load= and run=)
(Run SYSEDIT.EXE to gain access to the system files.)
Be sure there are no drivers loading in the Start-up folder.
Be sure the Palm Pilot autosync program is not being loaded at start-up.

* Please refer to Technote **A903** for BIOS/CMOS access keystrokes.

Legal Notice:

The information provided in the Rockwell Software Inc. (RSI) Knowledgebase is provided "as is" without warranty of any kind. RSI disclaims all warranties, either expressed or implied and in no event shall RSI be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of profit or damage, even if RSI have been advised of the possibility of such damages.

The foregoing limitation may not apply in those states that do not allow the exclusion or limitation of liability for damages.

Rockwell Software
Support Library

Receiving Unsolicited Messages in RSView32 with RSLinx

The information in this technote applies to:

Product:	**RSView32**	Type:	**Application Note**
Processor Family:		Technote ID:	**A449**
Package/Module(s):	**Works**	Modified:	**12/10/98**
Revision(s):	**5.01.26** **6.00.41**		
Fixed In Revision:			
Sub-System(s):	**RSLinx**		

Technical Note Details:

Background:
RSView32 does not directly support Unsolicited Messages. Both WinLinx and RSLinx support this feature when an Unsolicited DDE Topic for the device is configured. RSLinx version 1.7.62 or higher has a data collection mode that allows a DDE topic to be unsolicited with a scan rate. The maximum scan rate can be set to 3600000 msec (equivalent to 1 hour) allowing the user to display screens and immediately get the current value out of the PLC or SLC device and receive unsolicited messages when sent.

For More Information about the WinLinx Limitations:
Jump to Q260 - Receiving Unsolicited Messages in RSView with WinLinx

Configuration:

By configuring an Unsolicited DDE Topic in RSLinx, RSView32 can receive unsolicited writes to the tag database.

1. The appropriate RSLinx communication driver must be configured
2. Select Topic Configuration from the DDE menu
3. Enter a new topic name.
4. Click the Update List button to browse for the correct PLC or SLC device connection, and double click the device.
5. In the Data Collection Mode, make sure that both Poll Rate (msec) and Unsolicited Messages are checked.
6. We recommend setting the Poll Rate to its maximum value 3600000 msec (1 hour) for RSLinx 1.7, or a maximum value of 86400000 msec (24 hours) for RSLinx 2.0
7. Configure a DDE type of node in RSView32 using RSLinx as the Application and reference the Topic to the unsolicited topic name created in RSLinx.
8. Create a Device tag in the Tag Database in RSView32.
9. For the SLC5, be sure the address of the Device tag is the same as the Source File and Targets Destination File addresses as specified in the Message Function configuration in RSLogix 500.
10. For the PLC5, be sure the address of the Device tag is the same as the Data Table Addresses for This PLC5 and the Target device as specified in the Message Function configuration in RSLogix 5 for PLC5.

Message Configuration Details:

When a processor is generating an unsolicited write, there are 2 addresses specified. The first address indicates where the data is located locally, the second address indicates where the data is to be written to. When using Unsolicited and Polled DDE topics, both of these address MUST be the same.

How this works:
When a graphic is initially displayed, the data is read from the PLC5/SLC5. Thereafter, all updates are generated by the unsolicited message instruction in the PLC5/SLC5. Because the RSView32 / RSLinx application are initially reading the data, the source address must be equal to the target address. To guarantee the proper initialization of unsolicited data, the data must be polled once per hour.

Legal Notice:

Appendix B: OPC Specification Overview

OPC Overview

Version 1.0

October 27, 1998

Specification Type	Industry Standard Specification		
Title:	**OPC Overview**	Date:	October 27, 1998
Version:	1.0	Soft Source:	MS-Word Opcovw.doc
Author:	Opc Task Force	Status:	**Release**

Synopsis:

This specification serves as overview to OPC. It gives background information, motivation, architectural highlights and an abstract for each OPC topic.

NON-EXCLUSIVE LICENSE AGREEMENT

The OPC Foundation, a non-profit corporation (the "OPC Foundation"), has established a set of standard OLE/COM interface protocols intended to foster greater interoperability between automation/control applications, field systems/devices, and business/office applications in the process control industry.

The current OPC specifications, prototype software examples and related documentation (collectively, the "OPC Materials"), form a set of standard OLE/COM interface protocols based upon the functional requirements of Microsoft's OLE/COM technology. Such technology defines standard objects, methods, and properties for servers of real-time information like distributed process systems, programmable logic controllers, smart field devices and analyzers in order to communicate the information that such servers contain to standard OLE/COM compliant technologies enabled devices (e.g., servers, applications, etc.).

The OPC Foundation will grant to you (the "User"), whether an individual or legal entity, a license to use, and provide User with a copy of, the current version of the OPC Materials so long as User abides by the terms contained in this Non-Exclusive License Agreement ("Agreement"). If User does not agree to the terms and conditions contained in this Agreement, the OPC Materials may not be used, and all copies (in all formats) of such materials in User's possession must either be destroyed or returned to the OPC Foundation. By using the OPC Materials, User (including any employees and agents of User) agrees to be bound by the terms of this Agreement.

LICENSE GRANT:

Subject to the terms and conditions of this Agreement, the OPC Foundation hereby grants to User a non-exclusive, royalty-free, limited license to use, copy, display and distribute the OPC Materials in order to make, use, sell or otherwise distribute any products and/or product literature that are compliant with the standards included in the OPC Materials.

All copies of the OPC Materials made and/or distributed by User must include all copyright and other proprietary rights notices include on or in the copy of such materials provided to User by the OPC Foundation.

The OPC Foundation shall retain all right, title and interest (including, without limitation, the copyrights) in the OPC Materials, subject to the limited license granted to User under this Agreement.

WARRANTY AND LIABILITY DISCLAIMERS:

User acknowledges that the OPC Foundation has provided the OPC Materials for informational purposes only in order to help User understand Microsoft's OLE/COM

technology. THE OPC MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF PERFORMANCE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. USER BEARS ALL RISK RELATING TO QUALITY, DESIGN, USE AND PERFORMANCE OF THE OPC MATERIALS. The OPC Foundation and its members do not warrant that the OPC Materials, their design or their use will meet User's requirements, operate without interruption or be error free.

IN NO EVENT SHALL THE OPC FOUNDATION, ITS MEMBERS, OR ANY THIRD PARTY BE LIABLE FOR ANY COSTS, EXPENSES, LOSSES, DAMAGES (INCLUDING, BUT NOT LIMITED TO, DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL OR PUNITIVE DAMAGES) OR INJURIES INCURRED BY USER OR ANY THIRD PARTY AS A RESULT OF THIS AGREEMENT OR ANY USE OF THE OPC MATERIALS.

GENERAL PROVISIONS:

This Agreement and User's license to the OPC Materials shall be terminated (a) by User ceasing all use of the OPC Materials, (b) by User obtaining a superseding version of the OPC Materials, or (c) by the OPC Foundation, at its option, if User commits a material breach hereof. Upon any termination of this Agreement, User shall immediately cease all use of the OPC Materials, destroy all copies thereof then in its possession and take such other actions as the OPC Foundation may reasonably request to ensure that no copies of the OPC Materials licensed under this Agreement remain in its possession.

User shall not export or re-export the OPC Materials or any product produced directly by the use thereof to any person or destination that is not authorized to receive them under the export control laws and regulations of the United States.

The Software and Documentation are provided with Restricted Rights. Use, duplication or disclosure by the U.S. government is subject to restrictions as set forth in (a) this Agreement pursuant to DFARs 227.7202-3(a); (b) subparagraph (c)(1)(i) of the Rights in Technical Data and Computer Software clause at DFARs 252.227-7013; or (c) the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 subdivision (c)(1) and (2), as applicable. Contractor/ manufacturer is the OPC Foundation, P.O. Box 140524, Austin, Texas 78714-0524.

Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby.

This Agreement shall be governed by and construed under the laws of the State of Minnesota, excluding its choice or law rules.

This Agreement embodies the entire understanding between the parties with respect to, and supersedes any prior understanding or agreement (oral or written) relating to, the OPC Materials.

Table of Contents

Introduction

Readers Guide

This document serves as an overview to OPC. It gives background information, motivation, architectural highlights and an abstract for each OPC topic.
Specific interface specifications to develop OPC clients and/or OPC Servers (e.g., for DataAccess, Alarm&Event Handling or Historical DataAccess) and a specification for interfaces that are common for all OPC Servers are available as separate documents.

Chapter 1 gives some background information. It describes the purpose of OPC and why and how both vendors and customers have advantages in using OPC.
Chapter 2 provides a technical overview, describing the fundamentals of the design and characteristics of OPC components.
Chapter 3 describes the way OPC supports browsing of servers both locally and on remote machines.

OPC Background

A standard mechanism for communicating to numerous data sources, either devices on the factory floor, or a database in a control room is the motivation for OPC.
The information architecture for the Process Industry shown in Figure 0-1 involves the following levels:
Field Management. With the advent of "smart" field devices, a wealth of information can be provided concerning field devices that was not previously available. This information provides data on the health of a device, its configuration parameters, materials of construction, etc. All this information must be presented to the user, and any applications using it, in a consistent manner.
Process Management. The installation of Distributed Control Systems (DCS) and SCADA systems to monitor and control manufacturing processes makes data available electronically which had been gathered manually.
Business Management. Benefits can be gained by installing the control systems. This is accomplished by integrating the information collected from the process into the business systems managing the financial aspects of the manufacturing process. Providing this information in a consistent manner to client applications minimizes the effort required to provide this integration.
To do these things effectively, manufacturers need to access data from the plant floor and integrate it into their existing business systems. Manufacturers must be able to utilize off the shelf tools (SCADA Packages, Databases, spreadsheets, etc.) to assemble a system to meet their needs. The key is an open and effective communication architecture concentrating on data access, and not the types of data.

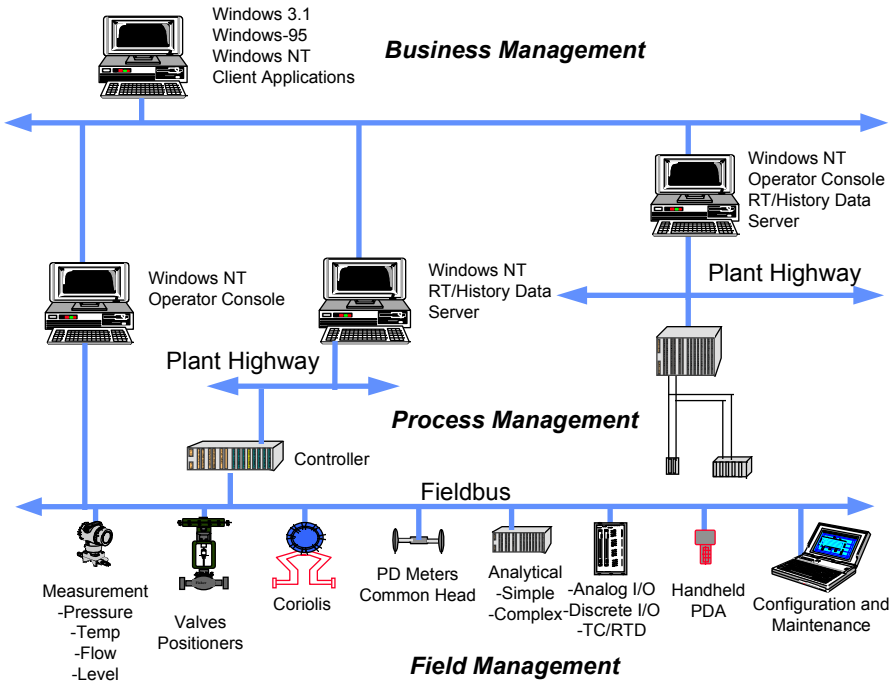


Figure 0-1 Process Control Information Architecture

Purpose

What is needed is a common way for applications to access data from any data source like a device or a database.

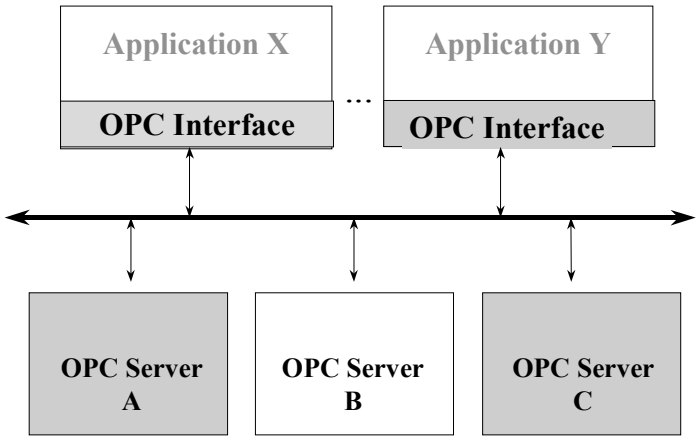


Figure 0-2. Applications Working with Many OPC Servers

OPC Server in this figure and in the following sections is used as synonym for any server that provides OPC interfaces, e.g.,

OPC DataAccess Server,

OPC Alarm&Event Server,
OPC HistoricalData Server.

The Current Client Application Architecture

There are many client applications that have been developed that require data from a data source and access that data by independently developing "Drivers" for their own packages.

This leads to the problems that follow:

- **Much duplication of effort**
  Everyone must write a driver for a particular vendor's hardware.
- **Inconsistencies between vendors drivers**
  Hardware features not supported by all driver developers.
- **Support for hardware feature changes**
  A change in the hardware's capabilities may break some drivers
- **Access Conflicts**
  Two packages generally cannot access the same device simultaneously since they each contain independent Drivers.

Hardware manufacturers attempt to resolve these problems by developing drivers, but are hindered by differences in client protocols. Today they cannot develop an efficient driver that can be used by all clients.

OLE for Process Control (OPC™) draws a line between hardware providers and software developers. It provides a mechanism to provide data from a data source and communicate the data to any client application in a standard way. A vendor can now develop a reusable, highly optimized server to communicate to the data source, and maintain the mechanism to access data from the data source/device efficiently. Providing the server with an OPC interface allows any client to access their devices.

The Custom Application Architecture

A growing number of custom applications are being developed in environments like Visual Basic (VB), Delphi, Power Builder, etc. OPC must take this trend into account. Microsoft understands this trend and designed OLE/COM to allow components (written in C and C++ by experts in a specific domain) to be utilized by a custom program (written in VB or Delphi for an entirely different domain). Developers will write software components in C and C++ to encapsulate the intricacies of accessing data from a device, so that business application developers can write code in VB that requests and utilizes plant floor data.

The intent of all specifications is to facilitate the development of OPC Servers in C and C++, and to facilitate development of OPC client applications in the language of choice.

The architecture and design of the interfaces are intended to support development of OPC servers in other languages as well.

General

OLE for Process Control (OPC™) is designed to allow client applications access to plant floor data in a consistent manner. With wide industry acceptance OPC will provide many benefits:

- Hardware manufacturers only have to make one set of software components for customers to utilize in their applications.
- Software developers will not have to rewrite drivers because of feature changes or additions in a new hardware release.
- Customers will have more choices with which to develop World Class integrated manufacturing systems.

With OPC, system integration in a heterogeneous computing environment will become simple. Leveraging OLE/COM the environment shown in Figure 0-3 becomes possible.

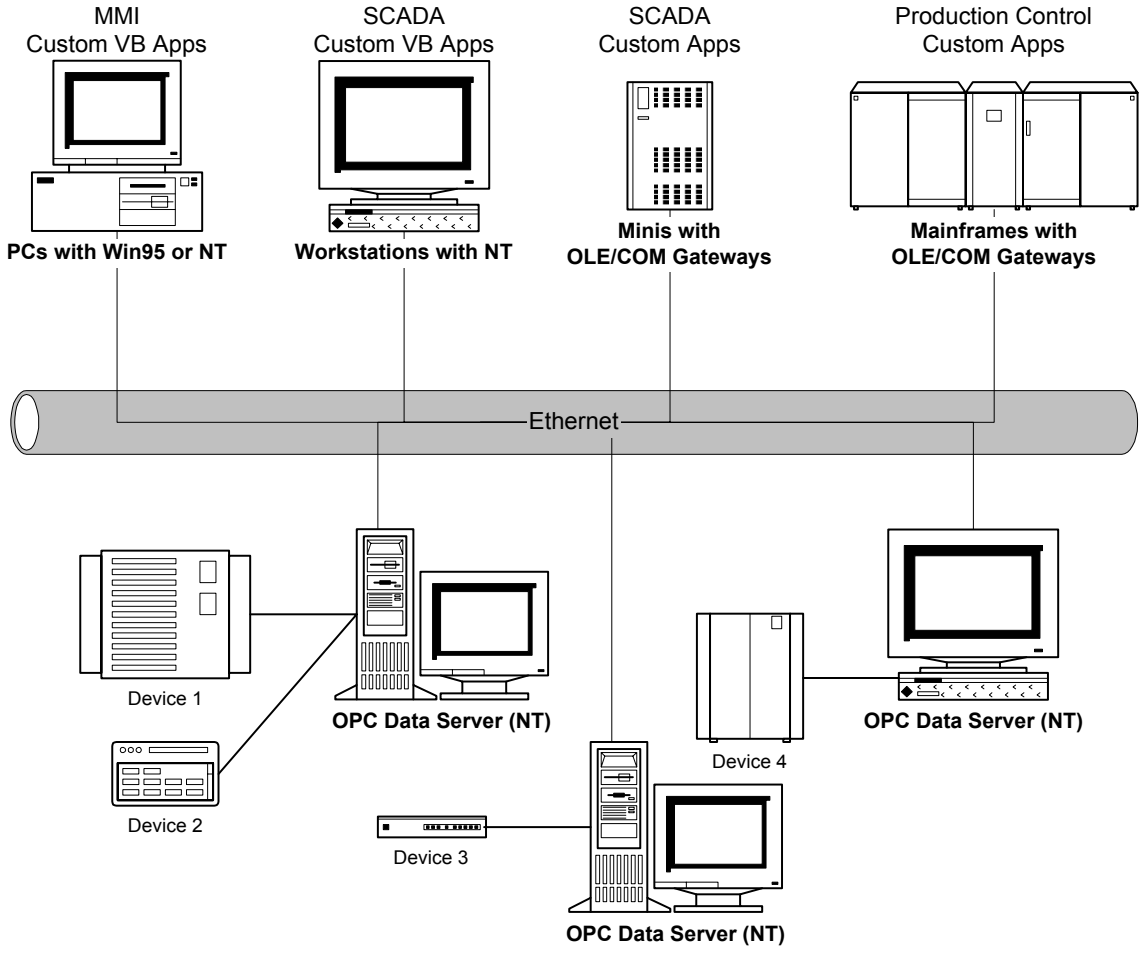


Figure 0-3. Heterogeneous Computing Environment

Scope

A primary goal for OPC is to deliver specifications to the industry as quickly as possible. With this in mind, the scope of the first document releases is limited to areas common to all vendors. Additional functionality will be defined in future releases. Therefore, the first releases focus on

- Online DataAccess, i.e., the efficient reading and writing of data between an application and a process control device flexibly and efficiently;
- Alarm and Event Handling, i.e., the mechanisms for OPC Clients to be notified of the occurrence of specified events and alarm conditions, and
- Historical Data Access, i.e., the reading, processing and editing of data of a historian engine.

Functionality such as security, batch and historical alarm and event data access belong to the features which are addressed in subsequent releases. The architecture of OPC leverages the advantages of the COM interface, which provides a convenient mechanism to extend the functionality of OPC.

Other goals for the design of OPC were as follows:

- simple to implement
- flexible to accommodate multiple vendor needs
- provide a high level of functionality
- allow for efficient operation

The specifications include the following:

1. A set of custom COM interfaces for use by client and server writers.
2. References to a set of OLE Automation interfaces to support clients developed with higher level business applications such as Excel, Visual Basic, etc.

The architecture is intended to utilize the Microsoft distributed OLE technology (DCOM) to facilitate clients interfacing to remote servers.

References

Kraig Brockschmidt ,Inside OLE, Second Edition, Microsoft Press, Redmond, WA, 1995.
Microsoft COM Specification, version 0.9, 10/24/95 (available from Microsoft's FTP site).
OLE Automation Programming Reference, Microsoft Press, Redmond, WA, 1996.
OLE 2 Programming Reference, Vol. 1, Microsoft Press, Redmond, WA, 1994.
The OPC Data Access Custom Specification 1.0A, OPC Foundation 1997.
The OPC Data Access Custom Specification 2.0, OPC Foundation 1998.
The OPC Data Access Automation Specification 2.0, OPC Foundation 1998.
The OPC Alarm and Event Access Specification 1.0, OPC Foundation 1998.
The OPC Historical Data Access Specification 1.0, OPC Foundation 1998.

OPC Fundamentals

OPC is based on Microsoft's OLE/COM technology.

OPC Objects and Interfaces

This specification describes the OPC COM Objects and their interfaces implemented by OPC Servers. An OPC Client can connect to OPC Servers provided by one or more vendors.

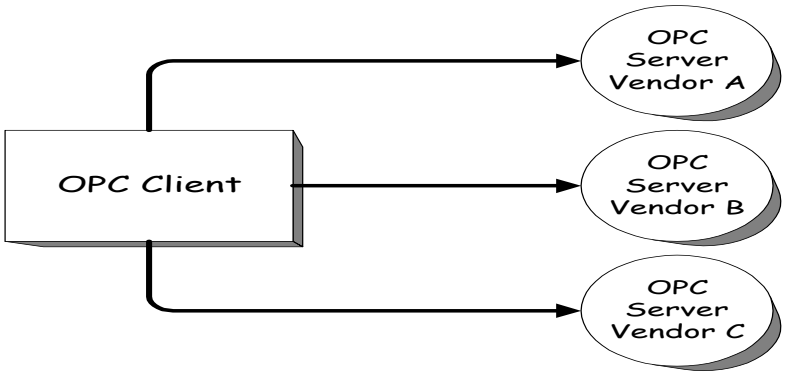


Figure 0-1 OPC Client

OPC Servers may be provided by different vendors. Vendor supplied code determines the devices and data to which each server has access, the data names, and the details about how the server physically accesses that data.

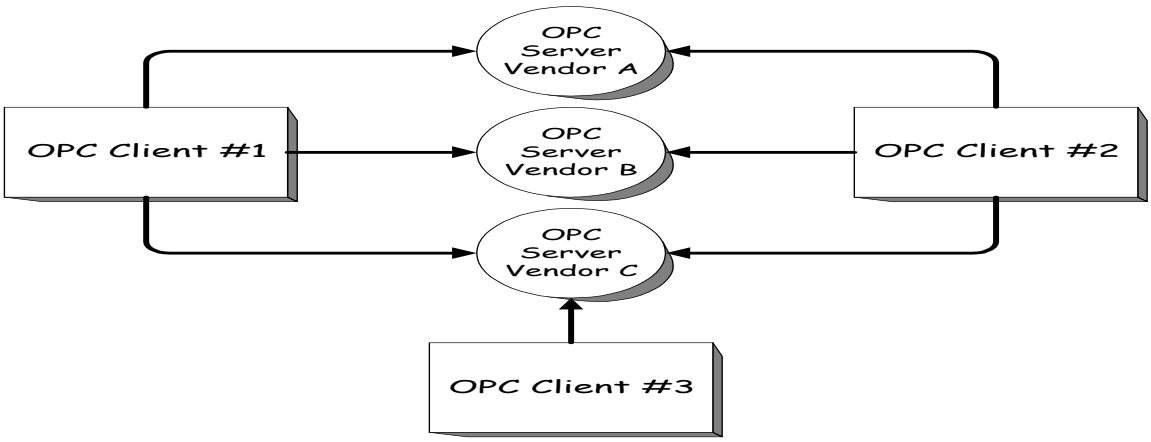


Figure 0-2 OPC Client/Server Relationship

OPC DataAccess Overview

At a high level, an OPC DataAccess Server is comprised of several objects: the server, the group, and the item. The OPC server object maintains information about the server and serves as a container for OPC group objects. The OPC group object maintains information about itself and provides the mechanism for containing and logically organizing OPC items.

The OPC Groups provide a way for clients to organize data. For example, the group might represent items in a particular operator display or report. Data can be read and written. Exception based connections can also be created between the client and the items in the group and can be enabled and disabled as needed. An OPC client can configure the rate that an OPC server should provide the data changes to the OPC client.

There are two types of groups, public and local (or 'private'). Public is for sharing across multiple clients, local is local to a client. Refer to the section on public groups for the intent, purpose, and functionality and for further details. There are also specific optional interfaces for the public groups.

Within each Group the client can define one or more OPC Items.

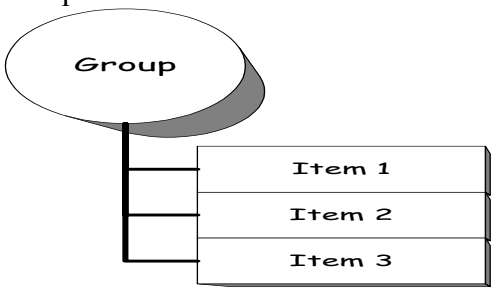


Figure 0-3 - Group/Item Relationship

The OPC Items represent connections to data sources within the server. An OPC Item, from the custom interface perspective, is not accessible as an object by an OPC Client. Therefore, there is no external interface defined for an OPC Item. All access to OPC Items is via an OPC Group object that "contains" the OPC item, or simply where the OPC Item is defined.

Associated with each item is a Value, Quality and Time Stamp. The value is in the form of a VARIANT, and the Quality is similar to that specified by Fieldbus.

Note that the items are not the data sources - they are just connections to them. For example, the tags in a DCS system exist regardless of whether an OPC client is currently accessing them. The OPC Item should be thought of as simply specifying the address of the data, not as the actual physical source of the data that the address references.

OPC Alarm and Event Handling Overview

These interfaces provide the mechanisms for OPC Clients to be notified of the occurrence of specified events and alarm conditions. They also provide services which allow OPC Clients to determine the events and conditions supported by an OPC Server, and to obtain their current status.

We make use of entities commonly referred to in the process control industry as *alarms* and *events*. In informal conversation, the terms *alarm* and *event* are often used interchangeably and their meanings are not distinct.

Within OPC, an *alarm* is an abnormal *condition* and is thus a special case of a *condition*. A *condition* is a named state of the OPC Event Server, or of one of its contained objects, which is of interest to its OPC Clients. For example, the tag FC101 may have the following conditions associated with it: HighAlarm, HighHighAlarm, Normal, LowAlarm, and LowLowAlarm.

On the other hand, an *event* is a detectable occurrence which is of significance to the OPC Server, the device it represents, and its OPC Clients. An event may or may not be associated with a condition. For example, the transitions into HighAlarm and Normal conditions are events which are associated with conditions. However, operator actions, system configuration changes, and system errors are examples of events which are not

related to specific conditions. OPC Clients may subscribe to be notified of the occurrence of specified events.

The IOPCEventServer interface provides methods enabling the OPC Client to:

- Determine the types of events which the OPC Server supports.
- Enter subscriptions to specified events, so that OPC Clients can receive notifications of their occurrences. Filters may be used to define a subset of desired events.
- Access and manipulate conditions implemented by the OPC Server.

In addition to the IOPCEventServer interface, an OPC Event Server may support optional interfaces for browsing conditions implemented by the server and for managing public condition groups (defined in the following section).

OPC Historical Data Access Overview

Historical engines today produce an added source of information that must be distributed to users and software clients that are interested in this information. Currently most historical systems use their own proprietary interfaces for dissemination of data. There is no capability to augment or use existing historical solutions with other capabilities in a plug-n-play environment. This requires the developer to recreate the same infrastructure for their products as all other vendors have had to develop independently with no interoperability with any other systems.

In keeping with the desire to integrate data at all levels of a business, historical information can be considered to be another type of data.

There are several types of Historian servers. Some key types supported by this specification are:

- Simple Trend data servers. These servers provided little else then simple raw data storage. (Data would typically be the types of data available from an OPC Data Access server, usually provided in the form of a tuple [Time Value & Quality])
- Complex data compression and analysis servers. These servers provide data compression as well as raw data storage. They are capable of providing summary data or data analysis functions, such as average values, minimums and maximums etc. They can support data updates and history of the updates. They can support storage of annotations along with the actual historical data storage.

Where OPC Fits

Although OPC is primarily designed for accessing data from a networked server, OPC interfaces can be used in many places within an application. At the lowest level they can get raw data from the physical devices into a SCADA or DCS, or from the SCADA or DCS system into the application.. The architecture and design makes it possible to construct an OPC Server which allows a client application to access data from many OPC Servers provided by many different OPC vendors running on different nodes via a single object.

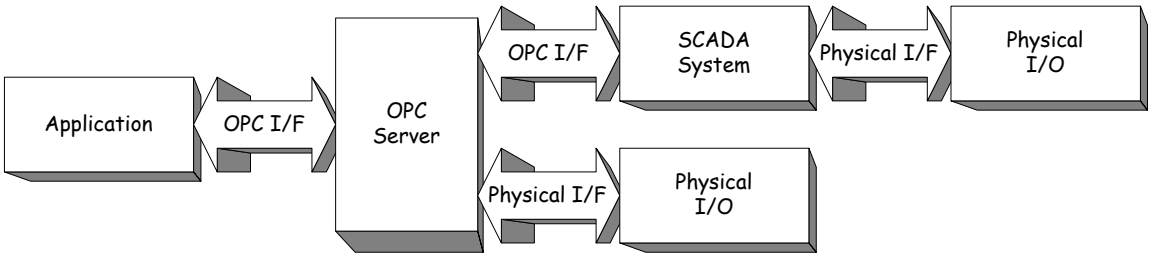


Figure 0-4 - OPC Client/Server Relationship

General OPC Architecture and Components

OPC specifications always contain two sets of interfaces; Custom Interfaces and Automation interfaces. This is shown in Figure 0-5.

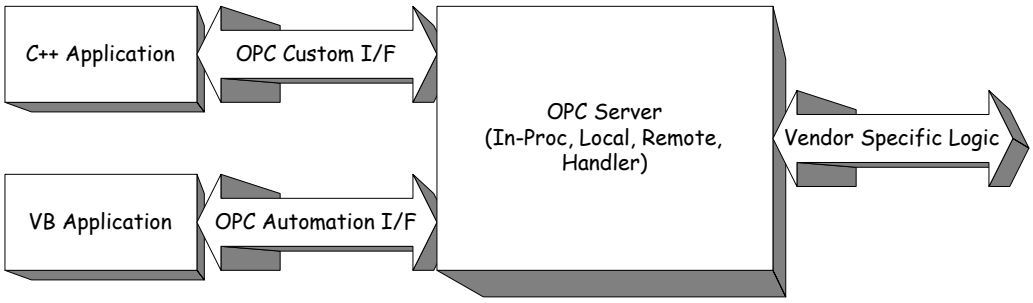


Figure 0-5 - The OPC Interfaces

The OPC Specification specifies COM interfaces (what the interfaces are), not the implementation (not the how of the implementation) of those interfaces. It specifies the behavior that the interfaces are expected to provide to the client applications that use them. Included are descriptions of architectures and interfaces that seemed most appropriate for those architectures. Like all COM implementations, the architecture of OPC is a client-server model where the OPC Server component provides an interface to the OPC objects and manages them.

There are several unique considerations in implementing an OPC Server. The main issue is the frequency of data transfer over non-sharable communications paths to physical devices or other data bases. Thus, we expect that OPC Servers will either be a local or remote EXE which includes code that is responsible for efficient data collection from a physical device or a data base.

An OPC client application communicates to an OPC server through the specified custom and automation interfaces. OPC servers must implement the custom interface, and optionally may implement the automation interface. In some cases the OPC Foundation provides a standard automation interface wrapper. This “wrapperDLL” can be used for any vendor-specific custom-server.

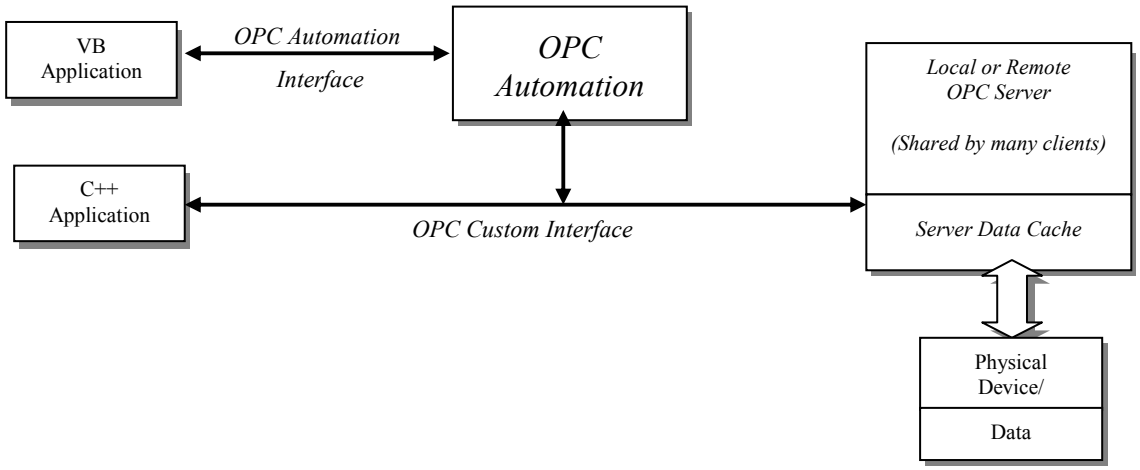


Figure 2-6 - Typical OPC Architecture

Local vs. Remote Servers

It is expected that OPC Server vendors will take one of two approaches to networking:

1. They can indicate that the client should always connect to a local server which makes use of an existing proprietary network scheme. This approach will commonly be used by vendors who are adding OPC capability to an existing distributed product.
2. They can indicate the client should connect to the desired server on the target node and make use of DCOM™ to provide networking. For this reason all of the RPC_E_* error codes should also be considered as possible returns from the functions below.

OPC Server Browser

The Interface of the OPC Server Browser (IOPCServerList) is specified as part of the OPCCommon document.

Overview of the Problem

The issue is, "How does a client program show the user what OPC Servers are available on a particular machine?". OPC Servers now register with the system via Component Categories. This allows the Microsoft ICatInformation (IID_ICatInformation) Interface on the StdComponentCatagoriesMgr (CLSID_StdComponentCategoriesMgr) to be used to determine which OPC servers are installed on the local machine. The problem is that this does not work for remote machines because the Component Categories Manager is a DLL and the ICatInformation interface only works in-process. As a result there is no easy way for a Client (including the Foundation supplied Automation Wrappers) to obtain a list of OPC Servers installed on a remote machine.

NOTE: the issue under discussion here is Server Browsing. This is entirely different from the Address Space browsing discussed in the OPC Data Access Interface.

Overview of the Solution

The OPC Foundation supplied Server Browser OPCENUM.EXE can reside on any machine, will access the local Component Categories Manger and provides a new interface IOPCServerList which can be marshaled and used by remote clients. This server has a published classid (see below) and can be installed once on any machine which hosts OPC servers. The client still needs to know the nodename of the target machine however he can now create this object remotely and use it's IOPCServerList interface to determine what types and brands of servers are available on that machine.

Appendix C: RSEmulate 5 & 500

RSEmulate 5 & 500

Table of Contents

Introduction to the Rockwell Software RSLogix Emulator

Rockwell Software's Emulator allows the testing of ladder logic and operator interface systems without the need for actual PLC hardware or I/O.

The function of the Emulator, simply put, is to exercise the large majority of ladder instructions in any ladder program without the need of any PLC related hardware.

The emulator does this by using the computer's CPU (instead of the CPU in a PLC) to scan the ladder logic. It takes your actual ladder logic (stored on disk in the .X5 or .ACH file) and actually scans the logic, executing each rung, testing for true/false conditions, and updating the data table.

Timers time, counters count, math instructions compute - the logic is executed as it would be in a real PLC-5 or SLC-500.

Under Windows, ***RSLogix Emulate 5*** or ***RSLogix Emulate 500*** are actively scanning the ladder logic and updating the data table. This allows other software programs such as RSLogix 500, RSTools, WINLogic 5, VIEW, RSTrend, etc, to act on and respond to a live ladder logic program.

Emulators available from Rockwell Software

This training guide discusses the concepts and techniques used in the emulation of ladder logic programs using Rockwell Software's RSLogix Emulate 5 or RSLogix Emulate 500 software.

Rockwell Software offers several options for emulation of ladder logic programs, both DOS and Windows based. This guide does not attempt to offer any detail on our DOS versions of emulation software, but the basic concepts for scanning ladder logic and simulating I/O are the same whether you are using Rockwell Software's Windows based emulators or the A.I. Series DOS based emulators.

RSLogix Emulation - Rockwell Software offers Windows based emulation packages for the PLC-5, *RSLogix Emulate 5*, and for the SLC-500, *RSLogix Emulate 500*. Note that both the A.I. Series PLC-5 and SLC-500 DOS based software program will function with the Windows based emulators.

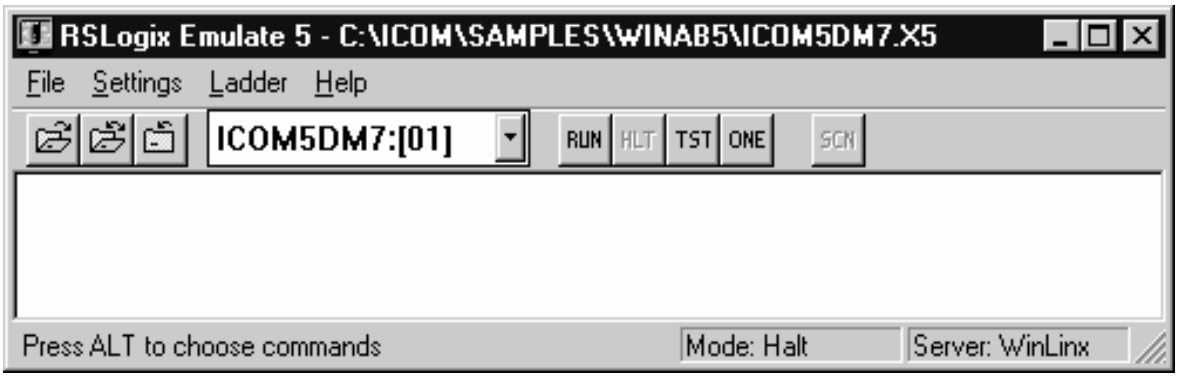
A.I. Series Emulation for DOS - A.I. Series DOS based offline emulation packages are available for the PLC-2, PLC-5, and SLC-500.

Setting Up and Basic Usage of the Emulator

Getting the Emulator to scan a normal ladder logic program is quite simple. You just turn it on! This involves two steps:

1. Tell the emulator which ladder logic file you want to scan. Do this via:
   File/Open and pick the ladder file (.X5 or .ACH).

2. Tell the emulator to start scanning. Do this via:
   Select either the *Run, Test* or *One Rung* scan modes (more on modes later).

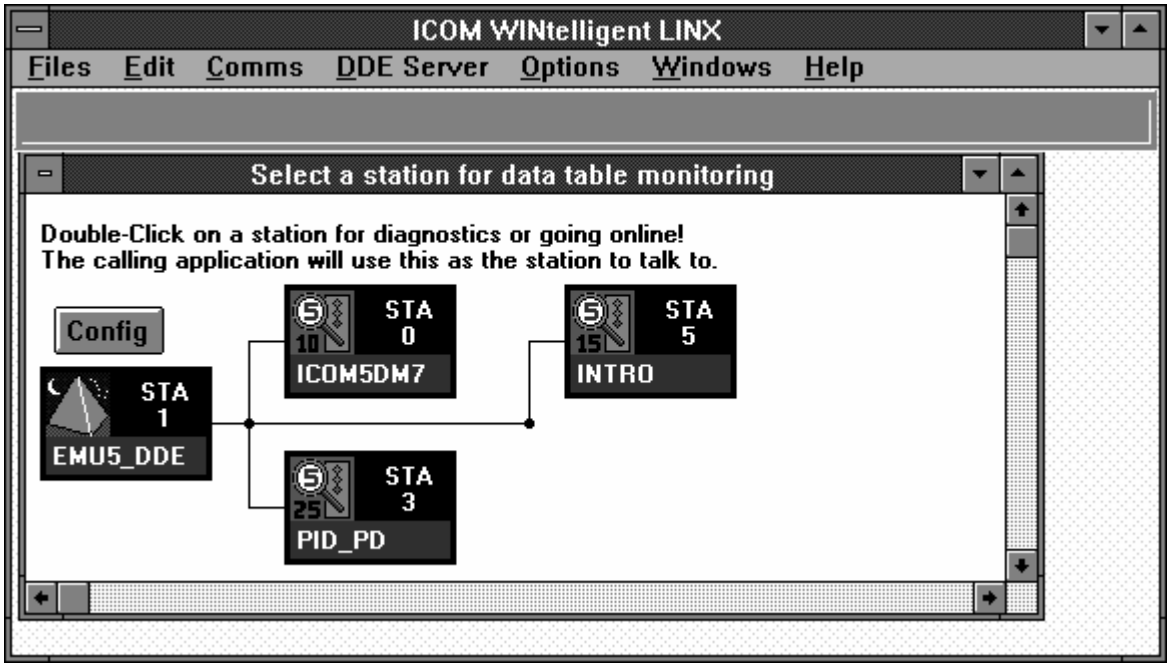


In the above example, a ladder logic file named ICOM5DM7.X5 is being emulated (note the path in the title bar). It is currently in the Halt mode (analogous to the program mode of a PLC).

The emulator can emulate multiple ladder logic files at one time, essentially creating a network of PLC's analogous to a DH+ or DH485.

The following "WHO" screen shows a WINtelligent LINX Super Who with multiple PLCs and the emulator.

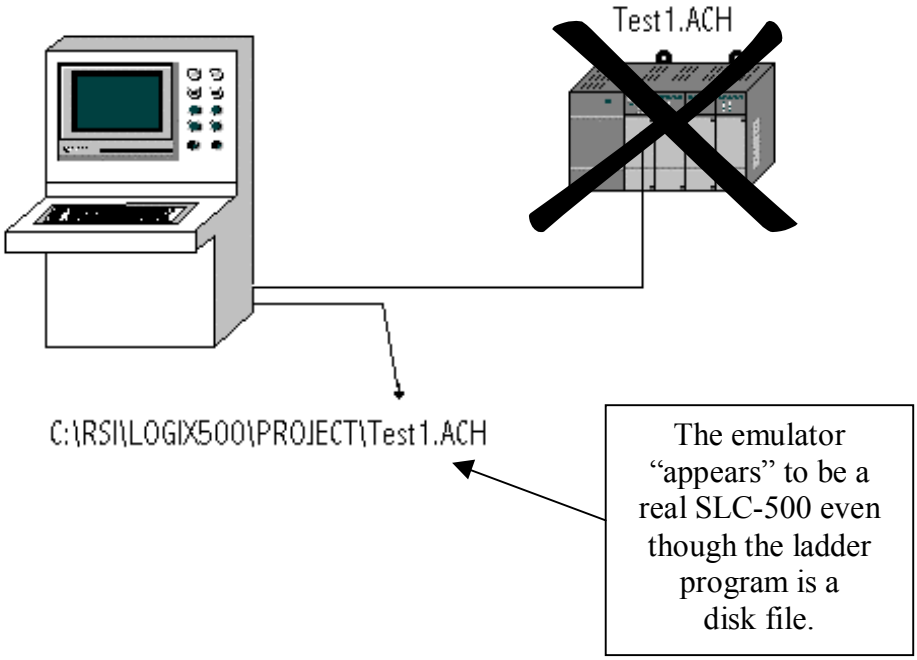
Note the "Emulator" is at station 1, the PLCs are at stations 0, 3 and 5 (PLC-5/10, 5/25, 5/15 respectively).



Using LINX with the Emulator

The Emulator uses RSLinx or Wintelligent LINX as its driver or "link" to other software packages such as RSView32 or RSTools HMI projects or RSLogix 500 or RSLogix 5 ladder logic software.

The emulator functions much the same as a real PLC. Under Windows, it runs separately, and does it's job (pretending to be a PLC) without being "aware" of what other software programs are "using" it. And from their perspective, other software packages such as VIEW or RSLogix aren't really "aware" that they are talking to an emulator rather than a PLC-5.



Linx is always required.

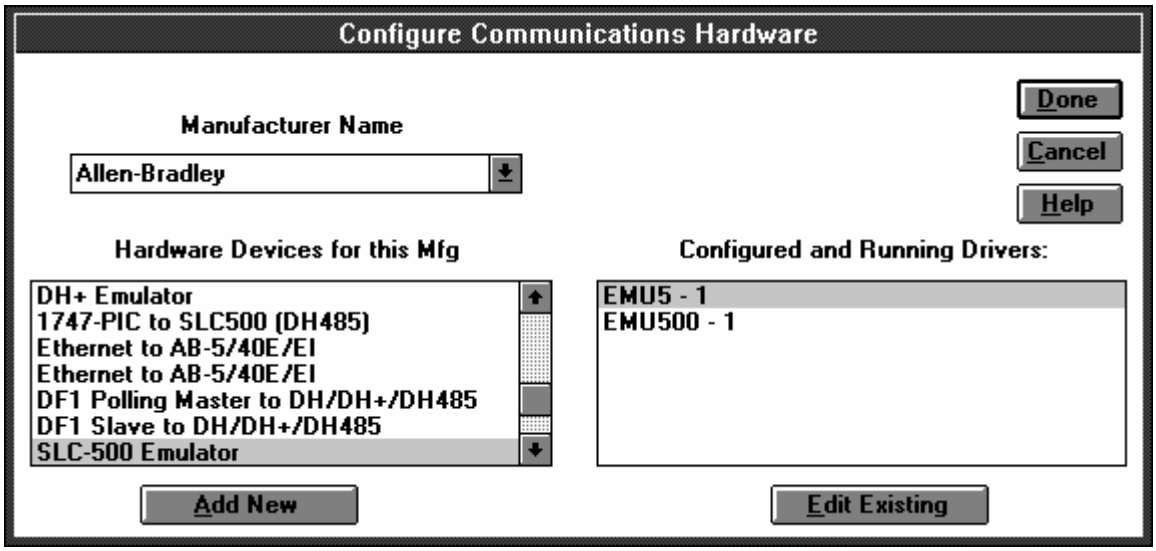
This situation is brought about by means of LINX being the "middle man" for Rockwell Software software under Windows, whether they are talking to a real PLC or an emulated PLC. In other words, from RSLogix 500's standpoint, it asks LINX to connect to a PLC - it really doesn't know or care if it's a real PLC or an emulated PLC.

RSLogix 500 ⇔ LINX ⇔ SLC-500

RSLogix 500 ⇔ LINX ⇔ RSLogix Emulate 500

Configuring LINX:

For software to communicate with a PLC (real or emulated), LINX must be running, and must have the proper communication driver set up and running. To add a communication driver in LINX (the procedure for RSLinx is very similar), from the Comms menu, choose Communication Hardware, and then choose the driver you require, in the case of the Emulator, either DH+ Emulator or SLC-500 Emulator as shown in the sample LINX screen below:



In the example above, both the PLC-5 emulator driver, EMU5, and the SLC-500 emulator driver, EMU500 are running. Note that LINX supports multiple drivers (analogous to multiple hardware networks such as DH+) and multiple PLC's on each highway. This is true for real or emulated PLCs.

“Simulating” real world conditions

Since the Emulator functions from within the computer environment, there needs to be some way of making the ladder program "think" that there is some real world I/O connected. The emulator allows for a variety of means of creating “action” in the emulated PLC, as well as offering several special troubleshooting tools you wouldn’t have with a real PLC.

Data Table

First of all, the emulated ladder logic program’s data table will act exactly as a real PLCs data table. So one option for simulating I/O and causing rung conditions to change is to write to the data table. This can be done using any means that one might normally use to write to the data table. For example, from LINX, one can write to the data table via the PLC Monitor option. Or from RSLogix 5, again, one could write data to the data table via all the normal means - one can simply write values to the data table, turn bits on or off, etc..

Or from an HMI such as RSView32 or WINtelligent VIEW, one has countless means of writing to the data table, for example, having some sort of download function from an operator interface screen. In other words, from an operator interface screen which would normally download a value to a PLC to change a timer preset, one could change the timer preset in the emulated PLC, and the ladder logic and data table would react accordingly.

Trouble shooting and “Debug” Tools

The actual writing of a ladder logic program is often only a small portion of the complete development cycle. The checkout and troubleshooting of a new program can involve considerably more time (and expense) than did the planning for and entry of the program.

The Rockwell Software Emulator was developed as a way to "work out the kinks" early in the program development cycle. Trapping program errors can be difficult at best. Often as the errors occur, data table values and the state of inputs or outputs change. In the process they invariably erase any of the clues that you would normally use to track down the problem.

Built into the Emulator are powerful "tools" to help isolate problems. These tools are discussed later in this guide.

Analog Emulation

Since the data table is fully emulated, you can simulate values raising and lowering if necessary by means of ladder logic, either in debug rungs (explained later), or simply rungs in the normal logic which would, via timers and math/compare/move instructions, raise and lower integer values in the data table. You could also use an HMI such as RSView32 or WINtelligent VIEW to change values, e.g., via slider bars or other downloading functions.

Although emulation of BTR/BTW (block transfer instruction) reads and writes is not supported (as there are no real world analog signals to read or write), there is limited emulation (DN and EN bits are set). PID and MSG instructions are not supported. For additional information, see the RSLogix Emulate 5 or 500 manual or the Help system section titled “What is and isn’t supported”.

Scanning Modes

The emulator allows for three types of ladder scanning modes and a Halt mode:

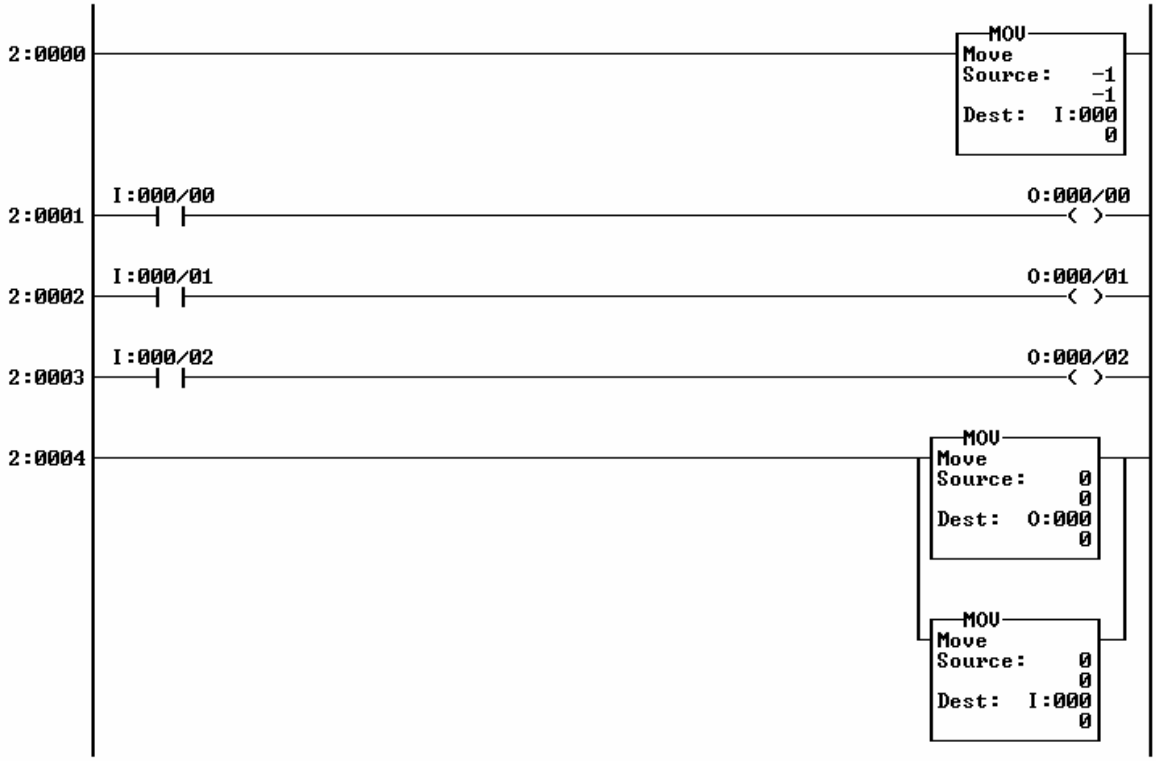
Run Mode: continuous scanning of the program from start to finish and over again.

Single/Test Mode: one complete scan of the program from start to finish.

One Rung Mode: (See sample below) Scanning one rung at a time based on user command.

Halt Mode: Somewhat analogous to the program mode, i.e. not scanning the program at all (but not actually a "Program" mode in that the Emulator does not support programming of ladder logic per se. You must make ladder logic changes in your programming software, save the changes, and then "reload" that ladder logic program into the Emulator).

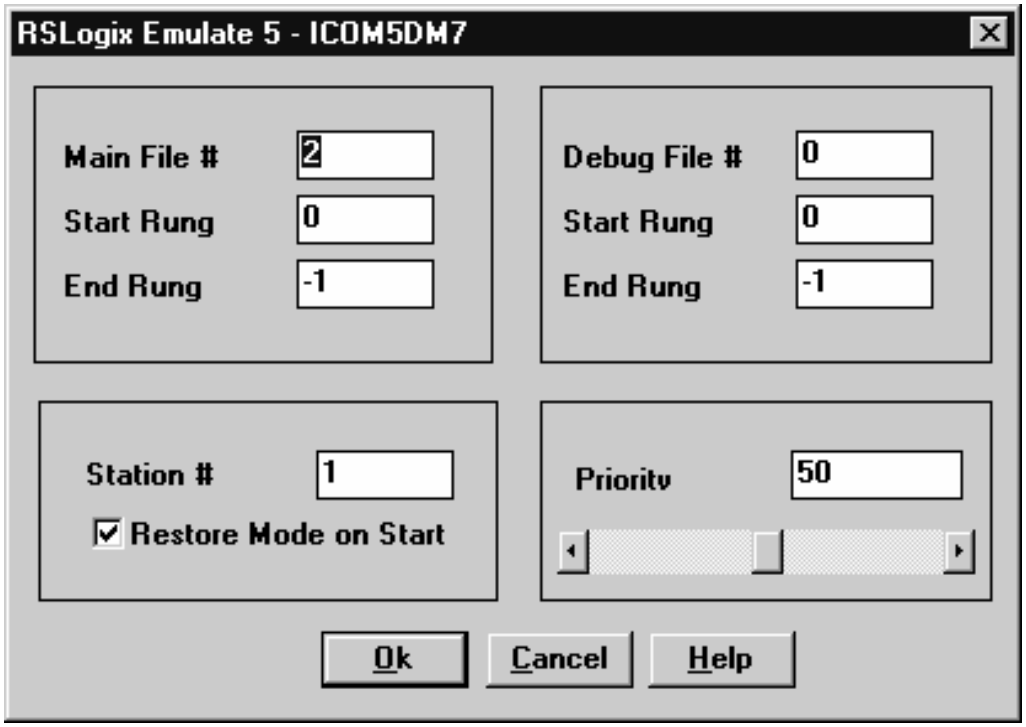
One Rung Mode Sample
The sample below is designed to test the One Rung Scan Mode. The first rung turns on the first input word. Rungs 1,2,3 each turn on one output bit as each rung is scanned. Rung 4 resets all the input and output bits. To see this in operation, arrange your desktop so that you can see RSLogix5 (online) and the emulator at the same time. Then choose the One button on the emulator, and then the Scan button several times to see the rungs execute.



Limit Execution to a Block of Rungs

The Emulator has the ability to selectively execute only the rungs in a block that you define. Rungs outside the block are ignored - they won't have any affect on the rungs under study. This comes in handy when trying to determine the reason why a section of logic isn't acting the way that you expected. Perhaps you have incorrectly entered an address elsewhere in the program that is overwriting the data that you expected to get.

Set the rungs you want to scan from the *Settings/Emulation* menu in the emulator as shown below:



Note that to have the emulator scan the entire program file, start rung should be 0 (zero) and end rung should be -1. Note that this is also where you would define a “debug file”.

BreakPoints

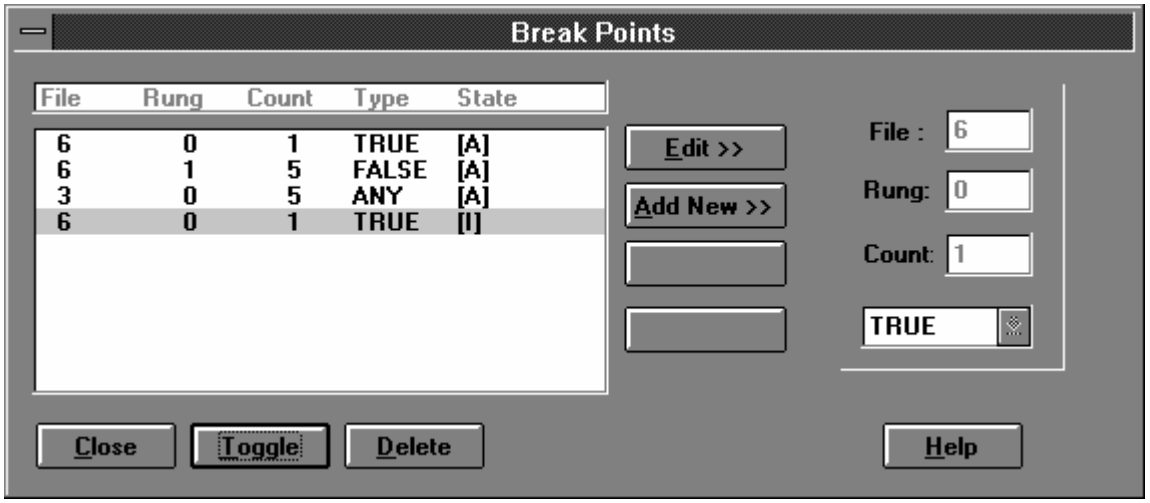
Breakpoints are special conditions that you assign to a rung or rungs. When these special conditions are met, scanning (execution) of your ladder logic will be frozen at that given rung/ breakpoint and the program scan will be turned to *One Rung Mode*. This way you can examine the data table, which is also now also frozen at that given point.

There are two circumstances where you may wish to assign breakpoints. Either you assign a breakpoint to an existing rung whose True/False continuity needs to be evaluated or you create a special rung(s) with conditions you want to capture.

For example, you might want to test for several inputs which should never go on at the same time. Or perhaps you might want to test for some data table condition where x = 2 and y = 3 (one or several data table addresses equal some particular value(s) that you want to know about).

You program the rungs in normal ladder logic (RSLogix or A.I. Series), but set the breakpoint conditions in the emulator itself. When the rung which has been assigned a breakpoint "hits" the breakpoint conditions, ladder logic scanning stops at that rung, and the emulator is in put into *One Rung Mode*.

The possible conditions for breakpoints are: True/False, or Any, and these interact with a counter so you can also include the number of times the rung has gone true, false or any (any simply means the rung has been scanned).



In the above example, there are four breakpoints, but the last one is inactive (and thus wouldn't have any effect). This might show a situation where, say, a special program file 6 was setup only to contain breakpoints, and in it are three special rungs which are designed to "trap" conditions. For example, the first breakpoint would "hit" if rung zero in program file six goes true once. The second breakpoint would hit if rung one in program file six goes false five times. The third breakpoint would "hit" if rung zero in program file 3 were

scanned at all, either true or false, five times (e.g., I could find out if that subroutine were being scanned).

Debug Files

Another special feature of the Emulator that functions to simulate I/O devices is the concept of "Debug" files. Debug files are ladder programs that you include as part of your application program. They are designated as debug program files which differentiates them from the more standard program files in one significant aspect, they are not transferred to the PLC processor during a download (and thus do not consume any actual PLC memory).

Developing and Using Debug Files

Operation of the Emulator does not necessarily require the use of debug files. Many means of causing action are available, as mentioned earlier, such as writing to the data table, using ladder logic such as recycling timers, or using an RSView32 HMI operator interface.

But debug files are a powerful tool for testing/troubleshooting your ladder logic programs. Used to "respond" to your normal ladder logic, a debug file is modeled after machine functions and is written in a familiar language - relay ladder logic.

As mentioned above, the primary difference between a normal ladder logic file and a debug ladder file is that the debug file is NOT transferred to the processor during a download. Otherwise, Debug files are much the same; they are written in ladder logic, access the same data table files and can "call" each other using the Jump to Subroutine (JSR) instructions.

While only one debug type file can be designated at any one time as the primary debug file, you can develop a number of program files that could later function as this primary file. The primary debug file is the one that will be used to actively simulate machine inputs. Numerous Debug files (up to the limit of the number of PLC Program files) can be programmed to permit you to develop the logic for a number of separate parts of the application that you plan to test. Maintaining a number of separate debug files means that you can switch rapidly between tests - there's no need to continuously modify a single file.

Multiple Debug files let you break-up the program such that smaller sections can be executed and checked out. Each section operates independent of the surrounding software. As an example, consider the situation where the "test" section contains inter-related math instructions. In this case, complex chains of instructions can be broken-up such that the intermediate results can be examined.

To completely separate one debug file from another (and from the rest of your normal logic), take advantage of the option to create new data files as "local" to a particular Debug

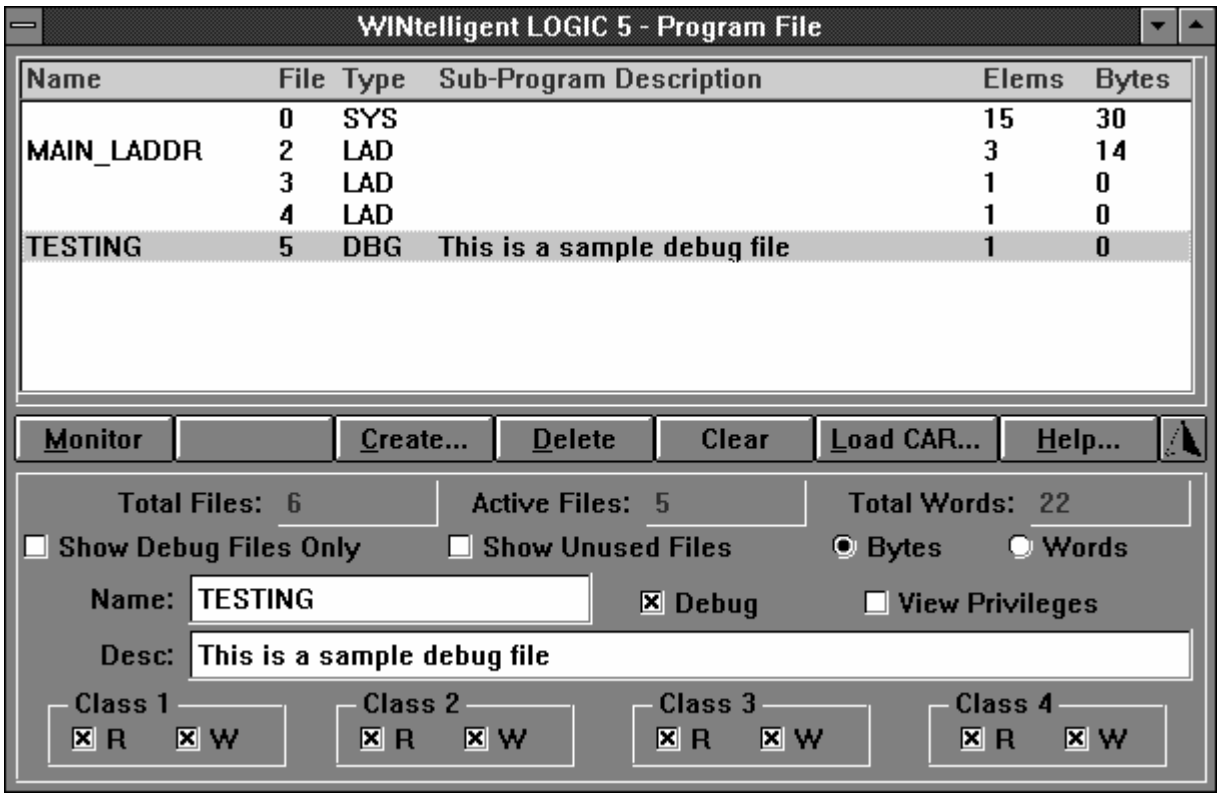
program file. That way, the information that the debug program temporarily stores cannot affect the normal flow of information in the rest of the data table.

In operation, the Emulator causes the primary debug file to be scanned and executed prior to scanning any normal (or non-debug) program files.

Creating a Debug File

To create a *Debug* program file, call up the Program File display (in A.I. Series or RSLogix) and create the new program file normally - just make sure you click the Debug check box. This will ensure this program file does not get downloaded to a real PLC, and also that the debug file gets scanned properly by the emulator if you have the emulator set to scan a debug file.

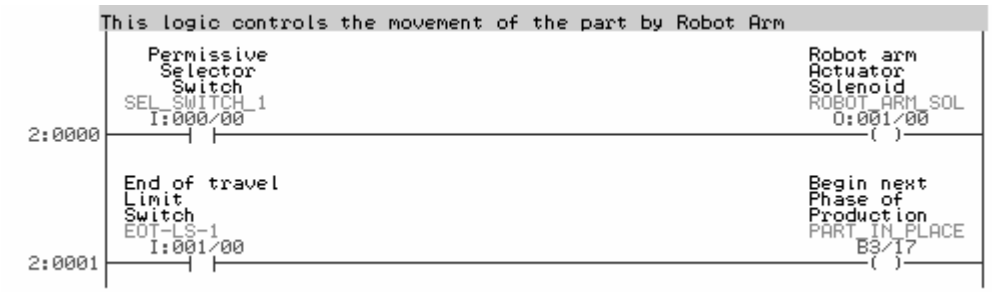
The following screen is an example of a WINtelligent LOGIC 5 Program File screen:



In the example above, the debug file is file 5. Once the debug file has been created, you can begin the process of developing the required ladder logic.

Example & Strategies for a Debug File

As a brief example, consider the following…In normal machine operation, to manipulate a part, an output from the programmable controller might open a solenoid which extends an arm attached to a hydraulic cylinder. As the arm reaches the end of its travel, a limit switch trips which de-energizes the output, which closes the valve and stops arm travel (if all goes well). Now that the arm is "in position" the machine continues through its cycle.



That would be the way that the system would operate if there was a PLC-5 processor, I/O racks, power supplies, a valve, etc.

With Emulation, checkout can be more convenient, and even more thorough (it can be done in your office instead of the shop floor). It does, however, require a rather thorough understanding of your machine operation since you must also build the debug files that will "respond" to the main ladder logic. This is far easier than it sounds because when you developed the original software, you probably had a type of script in mind. A typical script might have gone something like this:

"First, the output is energized which energizes the solenoid which starts the arm moving. The arm continues to move for a few seconds, finally tripping the end-of-travel limit switch. When the end-of-travel limit switch is tripped, the arm should stop and the heater should come on."

Now, as you begin developing the "Debug" file, which is what actually models the process, you visualize what sequence of events the machine would normally go through (as you did when you wrote the actual control logic). The Debug file in our example must recognize when the Robot Arm Solenoid output is energized, and introduce some delay while the arm would be extending, and then at the conclusion of that delay, turn on an input bit that simulates the limit switch closing.

The Debug file rungs, and the instructions that would be programmed on those rungs, are first described and then pictured on the following page:

Debug Rung #1

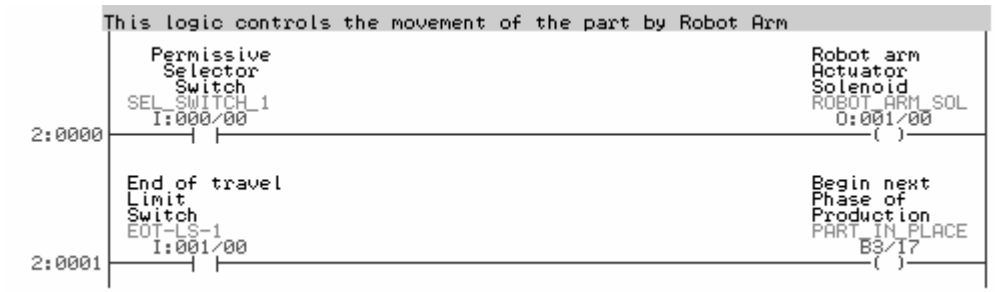
Purpose: Read the status of the solenoid output bit in the main program and then start a timer when true.
Instructions Used: Examine On (XIC) instruction with the same Symbol or address as the output connected to the solenoid.
Timer On Delay (TON) used to simulate the cylinder travel delay, programmed with an available timer address with a suitable time delay.

Debug Rung #2

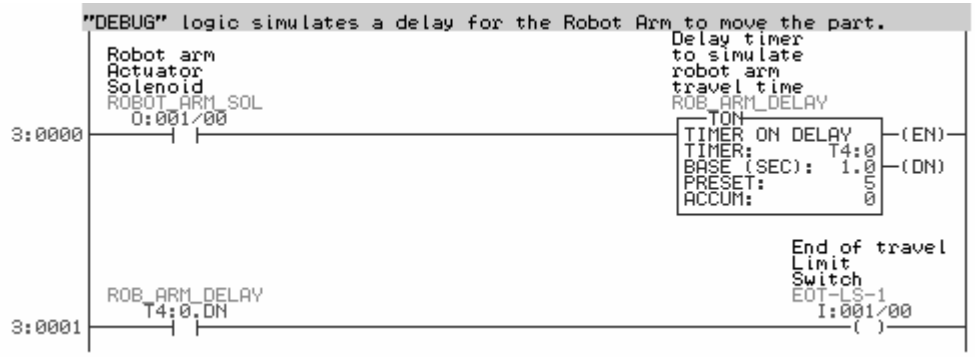
Purpose: To monitor the timers done bit and then simulate the limit switch closure.
Instructions Used: Examine On (XIC) instruction referencing the above timer's done bit.
Output Energize (OTE) instruction with the same Symbol or address as the input wired to the limit switch.

The rungs from both the normal and debug ladder logic would look like this:

First the normal rungs:

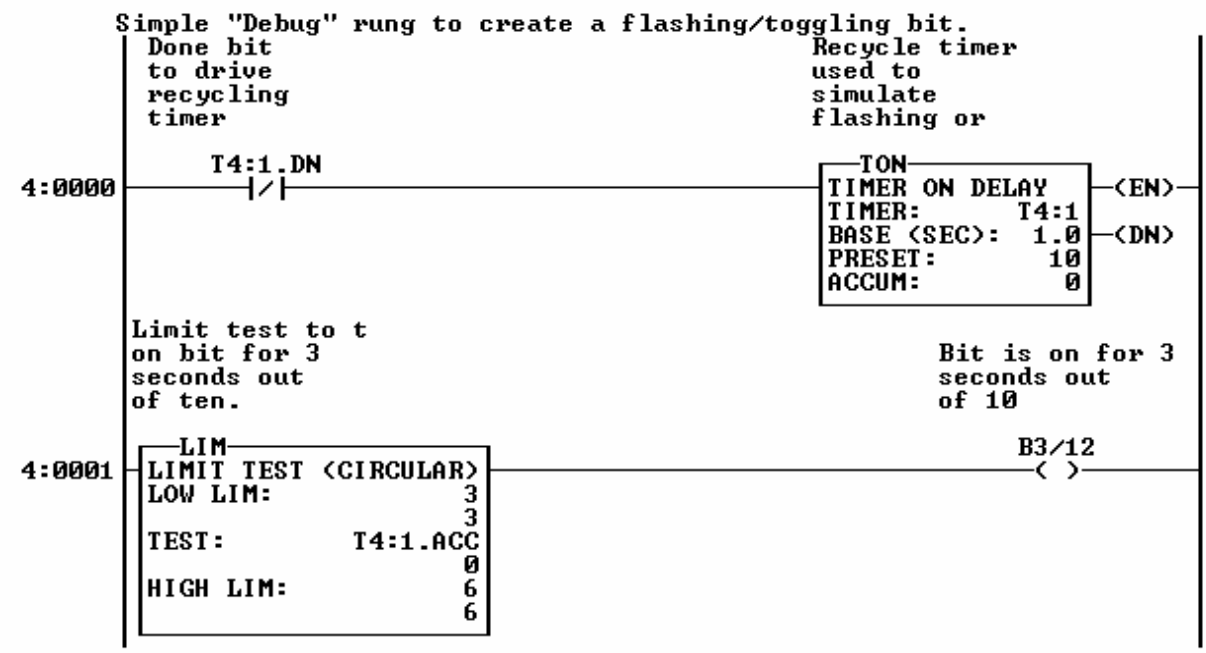


Next, the rungs from the debug file:



Another possibility for a debug file might be to simulate a bit toggling on/off every 3 seconds, or going on for 4 seconds every ten seconds, etc. Again, a recycling timer could be used, but this time in conjunction with a comparison instruction of some sort.

For example:



In the example above, B3/12 would come on for 4 seconds out of 10. To change the duration it is on, or the duration it is off, simply change the limits in the LIM instruction, or raise the accumulator in the Recycle Timer.

Appendix D: Using Network DDE

Network DDE:
Network DDE, referred to as NetDDE, is an extension of the DDE available in Windows. It provides information-sharing capabilities by opening two one-way pipes between applications. Network DDE is an extension of Dynamic Data Exchange (DDE) that can be used between two computers across the network.

Under Windows 95 or Windows NT, NetDDE allows a DDE client on one computer to read from and/or write to a DDE server on a different computer.

- For example, RSLinx could be running on one computer, and RSView32 or Excel could be running on a different computer.
  They could exchange data using DDE in much the same way as if they both resided on the same computer.

NetDDE takes all of the capabilities of DDE and extends this capability across the Windows network. NetDDE is not a special form of DDE; rather it involves a redirector that runs in the background on a Windows workstation searching for particular information contained in a DDE conversation.

One of the advantages of the implementation of NetDDE in Windows is that it requires no changes to any DDE-aware (compatible) applications (i.e., software programs that can communicate via DDE). All DDE-aware applications can take advantage of Network DDE.

***Note**: For Windows NT systems, **Network DDE Services** must be set up and started; they are not automatically started. You must have Windows NT administrator privileges to complete these procedures.*

*For a Windows 95 environment, you must manually start **NETDDE.EXE** before using Network DDE. Also, Rockwell Software is currently working on a utility to create DDE shares in Windows 95. For more information on this utility, contact Technical Support at 440-646-7801.*

***WARNING**: Do not use NetDDE for moderate to heavy data applications. It is best used to read a group of points every so often, i.e. end of shift reports.*

Starting Windows NT's NetDDE Services

Using NetDDE requires NT's NetDDE Services to be started. Windows NT Network DDE services are not automatically started. They must be set up and started.

You must have Windows NT administrator privileges to complete these procedures.

> ***Note**: If the client machine is running in a Windows 95 environment, you must manually start NETDDE.EXE before using Network DDE.*

1. Launch the Control Panel.
2. Double-click the Services icon.
3. Scroll down the list box and select Network DDE.
4. Click the Startup button to display a Service dialog box.
   - Optional: Click the Automatic radio button in the Startup Type section and click the OK button to return to the Services dialog box. This will cause NetDDE services to start automatically whenever NT starts.
5. Click the Services dialog box Start button.
6. Scroll down the list box and select Network DDE DSDM.
7. Click the Startup button to display a Service dialog box.
   - Optional: Click the Automatic radio button in the Startup Type section and click the OK button to return to the Services dialog box. This will cause NetDDE services to start automatically whenever NT starts.
8. If Services is not already running, click the Services dialog box Start button.
9. Click the Close button to exit the Services dialog box and return to Control Panel. You may close Control Panel as well.

Shared Topic Configuration:

With RSLinx, no user intervention is required to implement NetDDE. The only difference from the user's point of view is in the Application/Topic/Item string that they enter in the client application to implement DDE communications with RSLinx (see the examples on the next page).

But behind the scenes, in order for a Server application on one Windows NT or 95 computer to use NetDDE to communicate with a Client application running on another computer, a **DDE share** must first be created on the Server computer.

> ***Note**: RSLinx does not automatically create a NetDDE share entry in the NT or Windows 95 registry for configured topics. While Windows NT currently has the tools to create shares manually, Windows 95 does not offer this utility. Rockwell Software is currently working on a utility to create DDE shares in a Windows 95 machine acting as a server.*

Creating a Shared Topic in Windows NT:

Share entries are required in the Windows NT registry of the computer running the RSLinx Server for NetDDE to function. These shares must be created manually on the RSLinx Server using the DDESHARE utility that comes with NT.

1. Select the Run menu item to display a Run dialog box.
2. Type DDESHARE.EXE in the Command Line combo box. A DDE Shares window displays.
3. Select DDE Shares by double clicking the icon or select Shares menu item. A DDE Shares dialog box displays.
4. Click the Add a Share... button. A DDE Share Properties dialog box is displayed.
5. In the Share Name field, enter a name of your choice to remember this NetDDE share by. The name should be one that identifies RSLinx and the specific PLC and data available through the RSLinx topic you wish to use.
6. Enter the Application and Topic within the Static fields. For RSLinx, the application is RSLINX and the RSLinx topic you wish to use.
7. Check Allow start application and click the OK button to return to the DDE Share Properties dialog box.
8. Highlight the DDE share name you just entered and click the Trust Share button to display a Trusted Share Properties dialog box.
9. Check both the Start Application Enable and Initiate to Application Enable check boxes.
10. Click the Set button then click the OK button to return to the DDE Shares dialog box. Close the DDE Shares dialog box.

Syntax:

The client setup for entering NetDDE syntax is similar to standard DDE syntax. The major additional piece of information is that you must identify the target computer where the server (e.g. RSLinx) resides, and instead of using the Application/Topic, you use the ***share name*** (which identifies the application/topic)..

The computer name is added in the beginning of the command syntax, and the share name follows.

Typically, the NetDDE's client setup is as follows:

The **Application**(Service) field should contain:

\\COMPUTER NAME\NDDE$

[backslash] [backslash] [name of NetDDE server computer] [backslash][NDDE$]

The **Topic** field should contain:

Share Name

The **Item** field should contain the specific data item exactly as in normal DDE commands.

- In the case of RSLinx, it would be the PLC address, e.g., [N7:0].
- In the case of Excel, it would be the specific cell reference, e.g., [R1C1].

Examples:

Microsoft Excel 5.0 as Client, RSLinx as Server:

To establish a Network DDE link from Microsoft Excel 5.0 to RSLinx the syntax for the request is:

='\\COMPUTERNAME\NDDE$'|SHARENAME!'S:23'

The COMPUTERNAME would be the name of the computer that is running RSLinx (server). The computer name can be seen in the Network Properties window of NT.

The SHARENAME would be a valid share that has been created on the computer that is running RSLinx (server). The share name is created using the DDESHARE utility in NT.

RSTREND as Client, RSLinx as Server:

To establish a Network DDE hot link from RSTREND to RSLinx the syntax for the request is:

	Data Type	Application	Topic	Item	Length
1	DDE	\\COMPUTERNAME\NDDE$	RSLINX-TEST$	S:23	1

The **Application** is \\COMPUTERNAME\NDDE$. COMPUTERNAME is the name of the computer that has RSLinx installed on it and is to be the server of the data. This name is assigned in Windows NT.

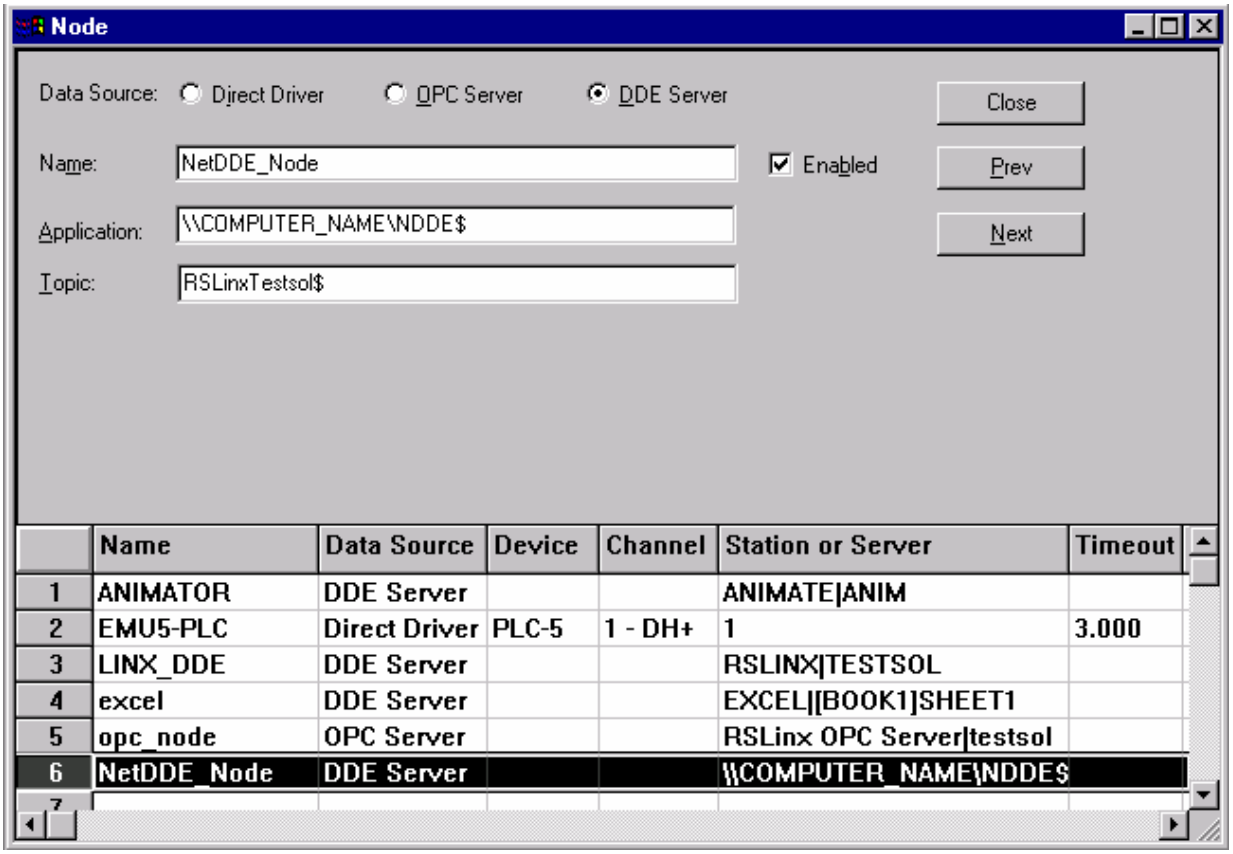
- To find this name start-up Control Panel in Windows and then double-click on the Network icon.

In this example, the **Sharename** is RSLinx-TEST$ as created in Windows NT using the DDESHARE.EXE utility.

The **Item** is S:23.

RSView32 as Client, RSLinx as the Server:

To establish a Network DDE link from RSView32 RSLinx create a DDE Node in RSView32's Node Editor:



	Name	Data Source	Device	Channel	Station or Server	Timeout
1	ANIMATOR	DDE Server			ANIMATE|ANIM	
2	EMU5-PLC	Direct Driver	PLC-5	1 - DH+	1	3.000
3	LINX_DDE	DDE Server			RSLINX|TESTSOL	
4	excel	DDE Server			EXCEL|[BOOK1]SHEET1	
5	opc_node	OPC Server			RSLinx OPC Server|testsol	
6	NetDDE_Node	DDE Server			\\COMPUTER_NAME\NDDE$	

In this example, the **Application** is \\COMPUTER_NAME\NDDE$. COMPUTER_NAME is the name of the computer that has RSView32 installed on it and is to be the server of the data. This name is assigned in Windows NT or 95.

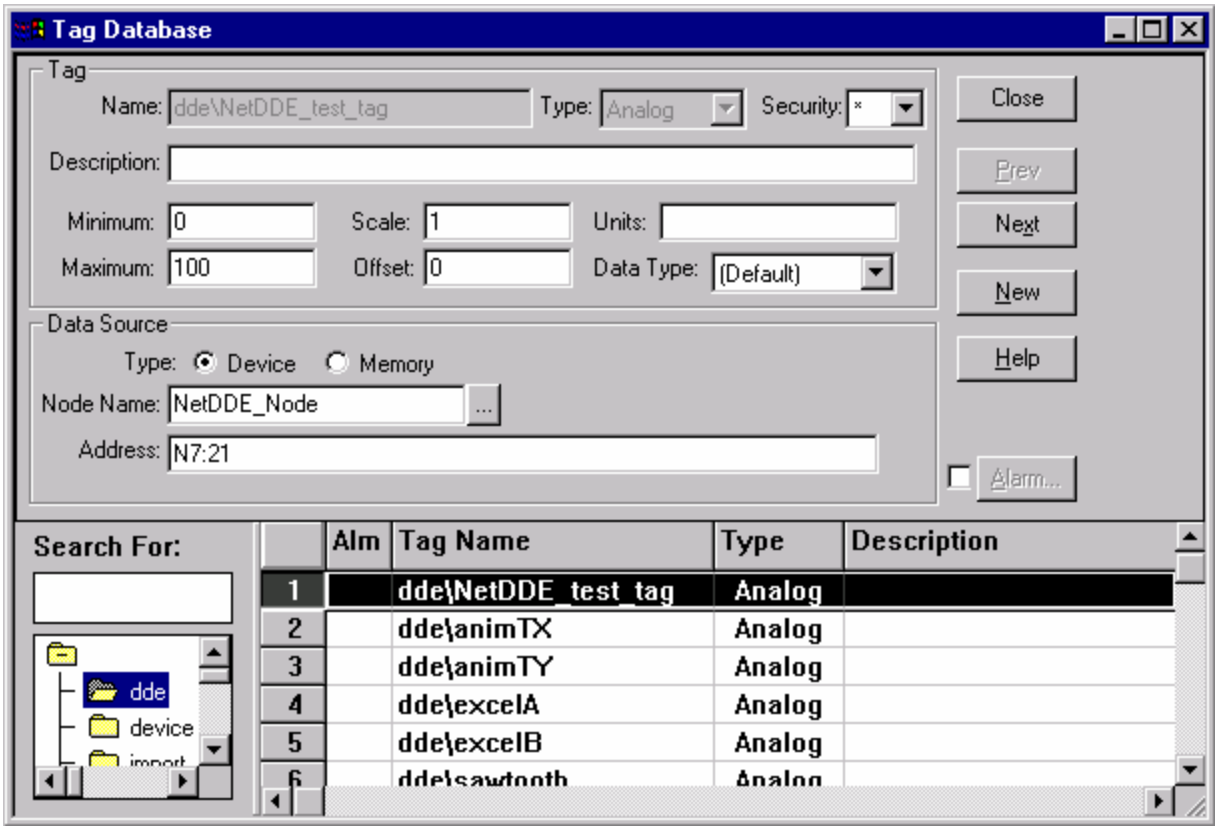
- To find your computer's name on the network, open the Control Panel in Windows and then double-click on the Network icon, then go to the Identification tab.

- To find the computer name of the computer you wish to connect to, double click the Network Neighborhood icon on the desktop and search the network for the computer name.

In this example at #6, the **Sharename** is RSLinxTESTSOL$.

The user would have setup the sharename in Windows NT by using the DDESHARE.EXE utility.

The **Item** is pictured below. It is created as a tag in RSView32's Tag Database Editor using the NetDDE node created earlier. In this case, the DDE item is the PLC address is N7:21.



In this Network DDE example, RSView32 would find the data for it's tag called "dde/NetDDE_test_tag" in:

- a computer on the network known as "COMPUTER_NAME"
- in that computer's RSLinx package and topic called TESTSOL as set up in the sharename
- The address of N7:21

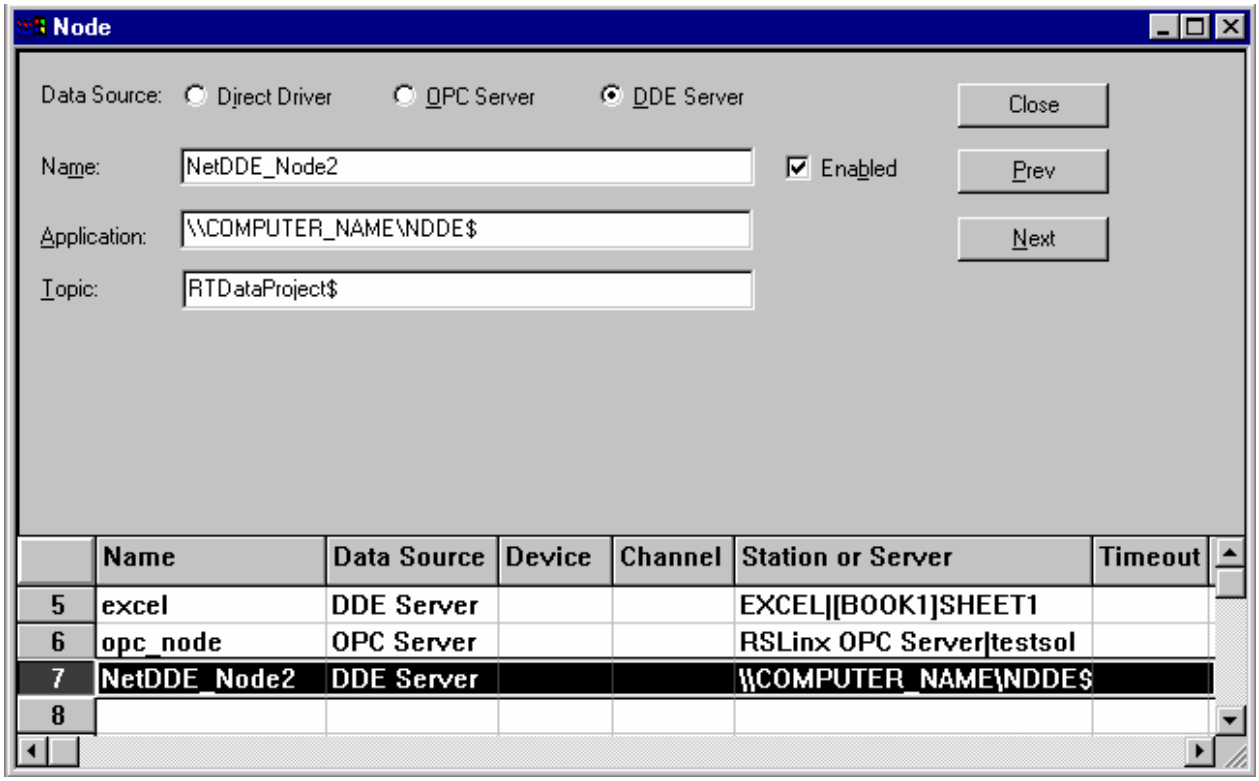
Beyond RSLinx - Another NetDDE Example

RSView32 as Client, Different RSView32 Computer as Server:

If you are running RSView32 on two different computers, both of which are networked, you can use Network DDE to share data between them.

When RSView32 acts as a NetDDE server, it provides real-time tag values to other RSView32 computers acting as NetDDE clients, or to any application that is NetDDE compliant. RSView32's DDE server supports a maximum of 3000 tags on scan.

To establish a Network DDE link from RSView32 to a different RSView32 machine on the network, create a DDE Node in RSView32's Node Editor:



Node #6 above called NetDDE_Node2 is a NetDDE node.

RSView32 compiles the application and topic together, separated by a pipe symbol (|), and it is placed the Station or Server field.

***Note**: RSView32 is a DDE client by default, which means that it will not share it's tag data with other DDE compliant programs unless configured to do so.*

- *A RSView32 command called* **RTDATASERVERON** *starts RSView32's DDE/OPC server and when the command is executed, RSView32 is changed to a Client/Server application.*

The **Application** is \\COMPUTER_NAME\NDDE$. COMPUTER_NAME is the name of the computer that has RSView32 installed on it and is to be the server of the data. This name is assigned in Windows NT or 95.

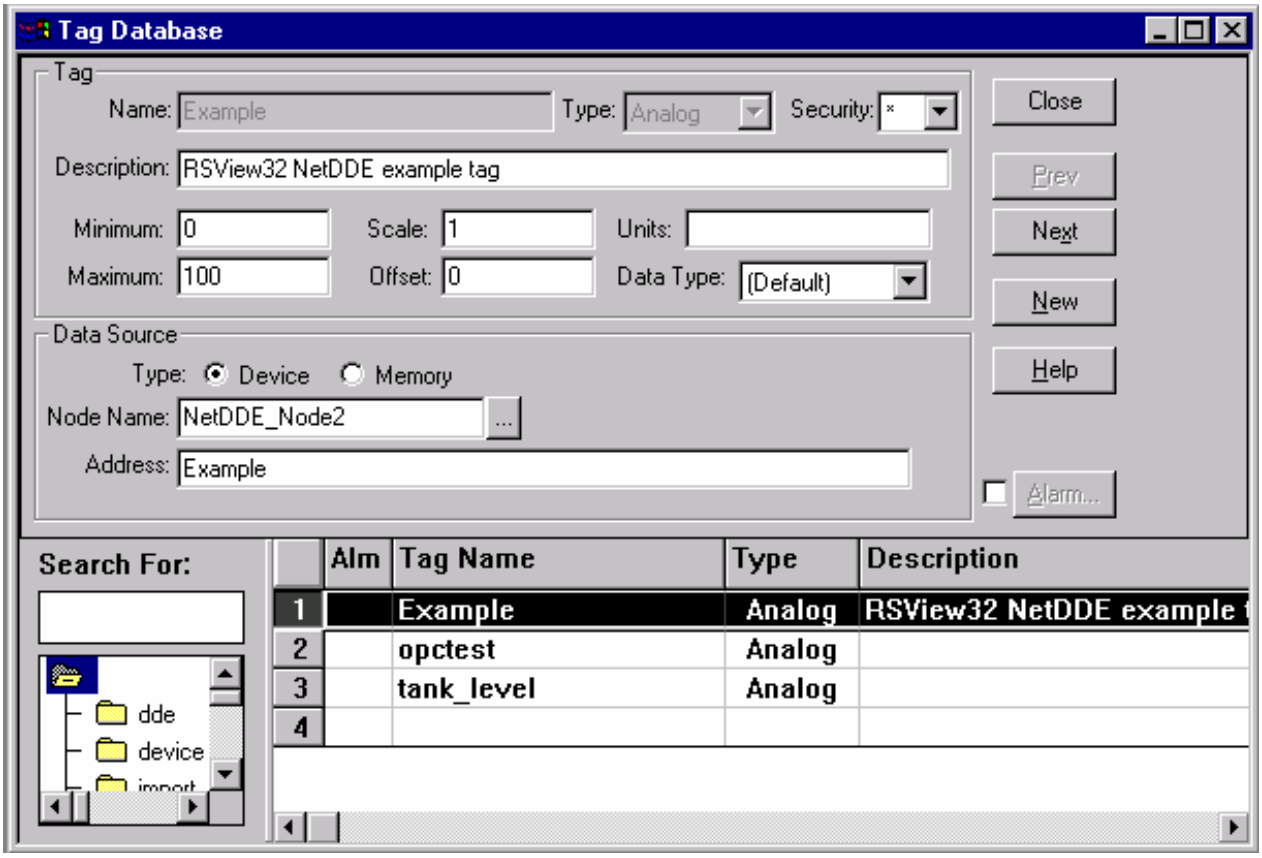
- To find your computer's name on the network, open the Control Panel in Windows and then double-click on the Network icon, then go to the Identification tab.
- To find the computer name of the computer you wish to connect to, double click the Network Neighborhood icon on the desktop.

In this example, the **Sharename** is RTdataProject$ and it goes in the Topic box.

- The user would have setup this sharename in Windows NT by using the DDESHARE.EXE utility.

***Note:** "RTData" is the application to use when referring to RSView32 as a DDE/OPC Server, and the project name that is running is to be used as Topic. This information would be necessary when configuring a DDE Share.*

The **Item** is pictured below. It is created as a tag in RSView32's Tag Database Editor using the NetDDE node created earlier. In this case, the DDE item is the RSView32 tag called EXAMPLE.



In this Network DDE example, RSView32 would find the data for this tag in:

- a computer on the network known as "COMPUTER_NAME"
- in that computer's RSView32 package and that RSView32 project called "PROJECT," as specified in the sharename.
- The address is an RSView32 tag called "EXAMPLE." from the RSView32 project named "PROJECT."

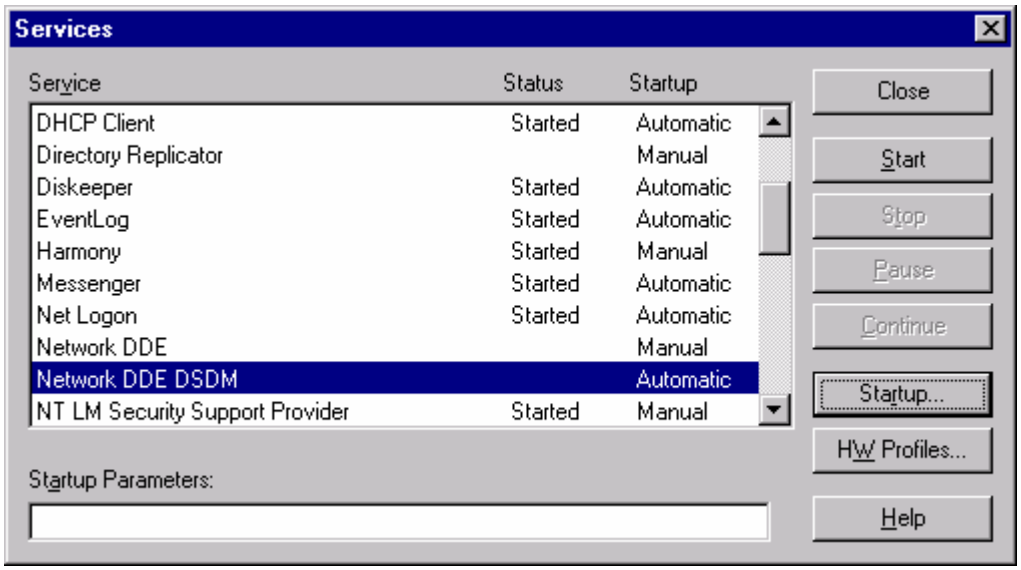
NetDDE Reminders

- If you are using NetDDE in Windows NT, you must configure the Network DDE DSDM services to start automatically before you can start the DDE Server in RSView32.

  This is configured under Services in the Control Panel. You only need t this configuration change once.

  - If you do not do this, an error message is reported. However, RSView32 can still function as a local DDE server. You only need to make this configuration change once.



- If you are using NetDDE on Windows 95, you must run NetDDE each time you start your computer by selecting Run from the Start menu and then typing NetDDE in the Open field.
  - You can start NetDDE automatically when you restart your computer by adding netdde.exe to your Startup folder.

www.rockwellautomation.com

Corporate Headquarters
Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions
Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36-BP 3A/B, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products
Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433
Europe: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 1774
Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 351 6723, Fax: (65) 355 1733

Publication RST-LNX-TGDE – February, 2002